schleicher
control systems

Betriebsanleitung

XCx 1100 / XCx 1200

Betriebsanleitung XCx 1100 / XCx 1200 • Version 01/14
Artikel-Nr. R4.322.2380.0 (322 385 78)

Zielgruppe

Die Betriebsanleitung ist für geschulte Fachkräfte ausgelegt. Es werden besondere Anforderungen an die Auswahl und Ausbildung des Personals gestellt, die mit dem Automatisierungssystem umgehen. Als Personen kommen z.B. Elektrofachkräfte und Elektroingenieure in Frage, die entsprechend geschult sind (siehe auch Sicherheitshinweise "Personalauswahl und -qualifikation").

Gültigkeit der Betriebsanleitung

ab Version Hardware Rev. 02 / Software V09.05/2

Vorgängerversion der Betriebsanleitung

12/12

Bezugsmöglichkeiten für Betriebsanleitungen

Alle Betriebsanleitungen können kostenlos vom Internet:
http://www.schleicher-electronic.com
geladen, oder unter Angabe der Artikel-Nr. bestellt werden bei:

SCHLEICHER Electronic
GmbH & Co. KG
Pichelswerderstraße 3-5
13597 Berlin

Germany

Zusätzliche Dokumentationen

Siehe Seite 10

Copyright by

SCHLEICHER Electronic
GmbH & Co. KG
Pichelswerderstraße 3-5
D-13597 Berlin

Germany

Telefon +49 30 33005-0
Telefax +49 30 33005-378
Hotline +49 30 33005-304
Internet *http://www.schleicher-electronic.com*
E-Mail info@schleicher-electronic.com

Änderungen und Irrtum vorbehalten

Inhaltsverzeichnis

1 Sicherheitshinweise 7
1.1 Bestimmungsgemäße Verwendung 7
1.2 Personalauswahl und -qualifikation 8
1.3 Projektierung, Programmierung, Installation, Inbetriebnahme und Betrieb 8
1.4 Gefahren durch elektrische Energie 9
1.5 Wartung und Instandhaltung 9
1.6 Umgang mit verbrauchten Batterien 9
2 Zusätzliche Betriebsanleitungen 10
3 Systemübersicht 11
3.1 Steuerungsaufbau 13
3.2 Montage 14
4 Steuereinheit XCA 1100 / XCA 1200 15
4.1 Schnittstellen, Bedienelemente, Anzeigen 16
4.1.1 X1, X2, X3 – Ethernet-Schnittstellen 17
4.1.2 X4, X5 – sercos III-Schnittstellen 17
4.1.3 X6/7, X8/9 – USB 2.0-Schnittstellen 17
4.1.4 X10 – DVI-Schnittstelle 18
4.1.5 X11 – CAN-Schnittstellen 19
4.1.6 X12 – RS 232 / RS 422 / RS 485-Schnittstellen 19
4.1.7 LED-Anzeigen Steuerung 20
4.2 LED-Anzeigen Ethernet und sercos III 21
4.2.1 Betriebsartenschalter 22
4.2.2 Reset-Taster 22
4.3 Technische Daten Steuereinheit XCA 23
4.4 Varianten XCA 1100 und XCA 1200 25
5 Erweiterungsmodule XCx 26
6 Inbetriebnahme 28
6.1 Installation von MULTIPROG, OPC-Server und AddOns 28
6.1.1 Systemvoraussetzungen 29
6.1.2 MULTIPROG installieren 30
6.1.3 OPC-Server installieren 30
6.1.4 AddOns installieren 31
6.1.5 Schleicher-Dialog installieren 32
6.2 Inbetriebnahme der Netzwerk-Schnittstelle 33
6.2.1 Vorbereitung 33
6.2.2 Kommunikationsschema 34
6.2.3 Computername für XCx vergeben 35
6.2.4 XCx-TCP/IP-Einstellungen anpassen 36
6.2.5 Kommunikation zur Programmierung mit MULTIPROG (Schritt 4) 39
6.3 Erste Schritte mit MULTIPROG 42
6.3.1 MULTIPROG starten, neues Projekt öffnen und speichern 42
6.3.2 Ein Projekt kompilieren und zur XCx senden 44
6.4 Einfügen des Koppelspeichers 46
6.4.1 Zugriff auf den Koppelspeicher 47
6.4.2 Hinweise zur Auswahl der Koppelspeicher-Version 48
6.5 Zugriff auf die I/O-Ebene 50
6.5.1 Die Buttons im Dialogfenster XUIO-Konfiguration 50
6.5.2 Einlesen der Hardware-Konfigurationen 52
6.5.3 Einstellen von Hardware-Konfigurations-Optionen 57
6.6 Zugriff auf Interrupt-Eingänge 59
6.6.1 Variablen für die Interruptverarbeitung 59
6.6.2 Beispiel-POEs für die Interruptverarbeitung 61
6.6.3 Taskstruktur für die Interruptverarbeitung 63
6.7 CANopen für dezentrale I/O 65
6.7.1 Spezifikationen 65
6.7.2 Anschlussprinzip und Verkabelung 66
6.7.3 Einstellungen am I/O-Modul RIO 8 I/O CANopen 67

6.7.4 Deklaration des I/O-Treibers für CANopen 67
6.7.5 Deklaration von Netzwerkvariablen in MULTIPROG 69
6.7.6 CANopen-Konfiguration mit "Schleicher CANopen Konfiguration" 70
6.7.7 CANopen-Konfiguration mit "ProCANopen" 71
6.7.8 Installation von ProCANopen 72
6.7.9 Einbindung von ProCANopen in MULTIPROG 73
6.7.10 Erste Verbindungen mit ProCANopen 74
6.8 Der Webserver 78
6.8.1 Allgemeine Funktionen und Konzept 78
6.8.2 Schleicher-spezifisches Applet 78
6.8.3 Deklaration von Variablen zur Visualisierung 78
6.8.4 Applikationsbeispiel 79
6.8.5 Browser / Komponenten 79
6.9 Allgemeine Hinweise zur Inbetriebnahme 79
7 Bedienung 80
7.1 Multi Function Application MFA 80
7.1.1 Start der MFA 80
7.1.2 Fensteraufbau der MFA 80
7.1.3 Funktionen der MFA 81
7.1.4 Die Log-Datei der MFA 83
7.1.5 Kommunikation mit anderen Applikationen 84
7.2 Schleicher-Dialog 85
7.2.1 Aufbau der Bedienoberfläche 85
7.2.2 Schleicher-Dialog SPS/CNC 86
7.2.3 Aufruf von Activ-Error-Buffer und Log-Book 89
8 Die SPS 90
8.1 Programmierung 90
8.2 SPS-Betriebszustände und Startverhalten 90
8.2.1 Betriebszustände 90
8.2.2 Wechseln der Betriebszustände mit MULTIPROG 91
8.2.3 Startverhalten der SPS nach dem Einschalten der Versorgungsspannung 92
8.3 Systemvariablen 93
8.4 Bibliotheken und Funktionsbausteine in MULTIPROG 94
8.4.1 Hinweis zu den Variablendeklarationen der Beispielprogramme von FBs 95
8.4.2 Bibliothek CANopen_Vxxx 96
8.4.3 Bibliothek CFB_Vxxx 96
8.4.4 Bibliothek CNC_Vxxx 97
8.4.5 Bibliothek Date_Time 98
8.4.6 Bibliothek MC_Vxxx 98
8.4.7 Bibliothek MMI 99
8.4.8 Bibliothek PLC_Vxxx 99
8.4.9 Bibliothek PNS_Vxxx 100
8.4.10 Bibliothek Profibus_Vxxx 100
8.4.11 Bibliothek SchleicherLib_Vxxx 100
8.4.12 Bibliothek Serial 100
8.4.13 Bibliothek XCx7_Vxxx 101
8.5 Das SPS-Betriebssystem ProConOS 102
8.5.1 Die Initialisierungsdatei ProConOS.INI 102
8.5.2 Beschreibung der ProConOS.INI Section- und Key-Einträge 102
9 Das Multi-Task-System 105
9.1 Übersicht 105
9.2 Anwender-Tasks 106
9.2.1 Zyklische Tasks 106
9.2.2 Ereignis-Tasks 107
9.2.3 System-Tasks 108
9.2.4 Default-Task 110
9.3 Anwender-Task-Information 111
9.4 Task-Prioritäten 113
9.5 Tasks und Watchdogs 114
9.6 Tasks einfügen und Programme zuweisen 115

10 Der Koppelspeicher 117
10.1 Variablen und Tasks 117
10.2 Zugriff auf den Koppelspeicher 118
10.3 Hilfe zum Koppelspeicher 118
10.4 Weitere Hintergrundinformationen zum Koppelspeicher 119
11 Die CNC 120
12 Serielle Schnittstelle RS232 121
12.1 Inbetriebnahme der seriellen Verbindung über die RS232-Schnittstelle 121
13 Weitere Betriebssoftware 123
13.1 Windows embedded 123
13.2 Schleicher X-Manager 123
13.2.1 "IP Configuration" 123
13.2.2 "EWF Configuration" 126
"NFS and Hardlink" 127
13.2.3 "RTOS Settings" 128
13.3 Remote Desktop UltraVNC 129
14 Anhang 130
14.1 Technische Daten aller Module 130
14.2 Zubehör und Software 131
14.3 Warenzeichenvermerke 131
14.4 Abbildungsverzeichnis 132
14.5 Tabellenverzeichnis 134
14.6 Index 135

Darstellungskonventionen

Sicherheits- und Handhabungshinweise werden in dieser Programmier-anleitung durch besondere Kennzeichnungen hervorgehoben:

Warnung!

Bedeutet, dass Personen, das Automatisierungssystem oder eine Sache beschädigt werden kann, wenn die entsprechenden Hinweise nicht eingehalten werden.

Kursivschrift: Hinweise zur Vermeidung der Gefährdung.

Wichtig! oder Hinweis

Hebt eine wichtige Information hervor, die die Handhabung des Automatisierungssystems oder den jeweiligen Teil der Betriebs-anleitung betrifft.

Weitere Objekte werden folgendermaßen dargestellt:

Objekt	Beispiel
Dateinamen	HANDBUCH.DOC
Menüs / Menüpunkte	*Einfügen / Grafik / Aus Datei*
Pfade / Verzeichnisse	*C:\Windows\System*
Hyperlinks	*http://www.schleicher-electronic.com*
Programmlisten	`MaxTsdr_9.6  = 60` `MaxTsdr_93.75 = 60`
Tasten	<Esc> <Enter> (nacheinander drücken) <Strg+Alt+Entf> (gleichzeitig drücken)
Bezeichner der Konfigurationsdaten	Q23
Namen von Variablen	*`mcMem.axSect[n].bContRel`*

1 Sicherheitshinweise

Der im folgenden verwendete Begriff Automatisierungssysteme umfasst Steuerungen sowie deren Komponenten (Module), andere Teile (wie z.B. Baugruppenträger, Verbindungskabel), Bediengeräte und Software, die für Programmierung, Inbetriebnahme und Betrieb der Steuerungen genutzt wird. Die vorliegende Betriebsanleitung kann nur einen Teil des Automatisierungssystems (z.B. Module) beschreiben.

Die technische Auslegung der Schleicher-Automatisierungssysteme basiert auf der Produktnorm EN 61131-2 (IEC 61131-2) für speicherprogrammierbare Steuerungen. Für die Systeme und Geräte gilt grundsätzlich die CE-Kennzeichnung nach der EMV-Richtlinie 2004/108/EG und sofern zutreffend auch nach der Niederspannungsrichtlinie 2006/95/EG.

Die Maschinenrichtlinie 98/37/EG bzw. 2006/42/EG ist nicht wirksam, da die in der Richtlinie genannten Schutzziele auch von der Niederspannungs- und EMV-Richtlinie abgedeckt werden.

Sind die Schleicher-Automatisierungssysteme Teil der elektrischen Ausrüstung einer Maschine, müssen sie vom Maschinenhersteller in das Verfahren zur Konformitätsbewertung einbezogen werden. Hierzu ist die Norm DIN EN 60204-1 zu beachten (Sicherheit von Maschinen, allgemeine Anforderungen an die elektrische Ausrüstung von Maschinen).

Von den Automatisierungssystemen gehen bei bestimmungsgemäßer Verwendung und ordnungsgemäßer Unterhaltung im Normalfall keine Gefahren in Bezug auf Sachschäden oder für die Gesundheit von Personen aus. Es können jedoch durch angeschlossene Stellelemente wie Motoren, Hydraulikaggregate usw. bei unsachgemäßer Projektierung, Installation, Wartung und Betrieb der gesamten Anlage oder Maschine, durch Nichtbeachten von Anweisungen in dieser Betriebsanleitung und bei Eingriffen durch ungenügend qualifiziertes Personal Gefahren entstehen.

1.1 Bestimmungsgemäße Verwendung

Die Automatisierungssysteme sind nach dem Stand der Technik und den anerkannten sicherheitstechnischen Regeln gebaut. Dennoch können bei ihrer Verwendung Gefahren für Leib und Leben des Benutzers oder Dritter bzw. Beeinträchtigungen von Maschinen, Anlagen oder anderen Sachwerten entstehen.

Das Automatisierungssystem darf nur in technisch einwandfreiem Zustand sowie bestimmungsgemäß, sicherheits- und gefahrenbewusst unter Beachtung der Betriebsanleitung benutzt werden. Der einwandfreie und sichere Betrieb der Steuerung setzt sachgemäßen Transport, sachgerechte Lagerung und Montage sowie sorgfältige Bedienung und Wartung voraus. Insbesondere Störungen, die die Sicherheit beeinträchtigen können, sind umgehend beseitigen zu lassen.

Die Automatisierungssysteme sind ausschließlich zur Steuerung von Maschinen und Anlagen vorgesehen. Eine andere oder darüber hinausgehende Benutzung gilt nicht als bestimmungsgemäß. Für daraus resultierende Schäden haftet der Hersteller nicht.

Zur bestimmungsgemäßen Verwendung der Automatisierungssysteme sind die in dieser Betriebsanleitung beschriebenen Anweisungen zum mechanischen und elektrischen Aufbau, zur Inbetriebnahme und zum Betrieb zu beachten.

1.2 Personalauswahl und -qualifikation

Wichtig!

Alle Projektierungs-, Programmier-, Installations-, Inbetriebnahme-, Betriebs- und Wartungsarbeiten in Verbindung mit dem Automatisierungssystem dürfen nur von geschultem Personal ausgeführt werden (z.B. Elektrofachkräfte, Elektroingenieure). Das Projektierungs- und Programmierpersonal muss mit den Sicherheitskonzepten der Automatisierungstechnik vertraut sein.

Das Bedienpersonal muss im Umgang mit der Steuerung unterwiesen sein und die Bedienungsanweisungen kennen.

Das Installations-, Inbetriebnahme- und Wartungspersonal muss eine Ausbildung besitzen, die zu Eingriffen am Automatisierungssystem berechtigt.

1.3 Projektierung, Programmierung, Installation, Inbetriebnahme und Betrieb

Das Automatisierungssystem ist in seiner Anwendung zumeist Bestandteil größerer Systeme oder Anlagen, in denen Maschinen gesteuert werden. Bei Projektierung, Installation und Inbetriebnahme der Automatisierungssysteme im Rahmen der Steuerung von Maschinen müssen deshalb durch den Maschinenhersteller und Anwender die Sicherheitsbestimmungen der Maschinenrichtlinie 98/37/EG bzw. 2006/42/EG beachtet werden. Im spezifischen Einsatzfall geltende nationale Unfallverhütungsvorschriften wie z.B. VBG 4.0.

Alle sicherheitstechnischen Vorrichtungen der gesteuerten Maschine sind so auszuführen, dass sie unabhängig von der Steuerung funktionieren. Not-Aus-Einrichtungen müssen in allen Betriebsarten der Steuerung wirksam bleiben. Im Not-Aus-Fall müssen die Versorgungsspannungen aller von der Steuerung angesteuerten Schaltelemente in einen sicheren Zustand gebracht werden.

Es sind Vorkehrungen zu treffen, dass nach Spannungseinbrüchen und -ausfällen ein unterbrochenes Steuerungsprogramm ordnungsgemäß wieder aufgenommen werden kann. Dabei dürfen auch kurzzeitig keine gefährlichen Betriebszustände auftreten. Gegebenenfalls ist Not-Aus zu erzwingen.

Damit ein Leitungsbruch auf der Signalseite nicht zu undefinierten Zuständen in der Steuerung führen kann, sind bei der E/A-Kopplung hard- und softwareseitig entsprechende Sicherheitsvorkehrungen zu treffen. Einrichtungen der Steuerungstechnik und deren Bedienelemente sind so einzubauen, dass sie gegen unbeabsichtigte Betätigung ausreichend geschützt sind.

1.4 Gefahren durch elektrische Energie

Warnung!

Nach Öffnen des Systemschrankes oder nach Entfernen des Gehäuses von Systemkomponenten werden bestimmte Teile des Automatisierungssystems zugänglich, die unter gefährlicher Spannung stehen können.

Die Spannung abschalten, bevor an den Geräten gearbeitet wird. Bei Messungen unter Spannung Kurzschluss vermeiden.

Der Anwender muss dafür sorgen, dass unbefugte und unsachgemäße Eingriffe unterbunden werden (z.B. verschlossener Schaltschrank).

Das Personal muss gründlich mit allen Gefahrenquellen und Maßnahmen zur Inbetriebnahme und Wartung gemäß den Angaben in der Betriebsanleitung vertraut sein.

1.5 Wartung und Instandhaltung

Werden Mess- oder Prüfarbeiten am aktiven Gerät erforderlich, dann sind die Festlegungen und Durchführungsanweisungen der nationalen Unfallverhütungsvorschriften, wie z.B. VBG 4.0, zu beachten. Es ist geeignetes Elektrowerkzeug zu verwenden.

Reparaturen an Steuerungskomponenten dürfen nur von autorisierten Reparaturstellen vorgenommen werden. Unbefugtes Öffnen und unsachgemäße Eingriffe oder Reparaturen können zu Körperverletzungen oder Sachschäden führen.

Vor Öffnen des Gerätes ist immer die Verbindung zum speisenden Netz zu trennen (Netzstecker ziehen oder Trennschalter öffnen).

Steuerungsmodule dürfen nur im spannungslosen Zustand gewechselt werden. Demontage und Montage sind gemäß den mechanischen Aufbaurichtlinien vorzunehmen.

Beim Auswechseln von Sicherungen dürfen nur Typen verwendet werden, die in den technischen Daten spezifiziert sind.

Beim Austausch von Batterien dürfen nur Typen verwendet werden, die in den technischen Daten spezifiziert sind. Batterien sind in jedem Fall nur als Sondermüll zu entsorgen.

1.6 Umgang mit verbrauchten Batterien

Die in den Automatisierungssystemen verwendeten Batterien sind, nach deren Gebrauchsende, dem Gemeinsamen Rücknahmesystem Batterien (GRS) oder öffentlich-rechtlichen Entsorgungsträgern zuzuführen.

Batterien sollen nur im entladenen Zustand zurückgegeben werden. Der entladene Zustand ist erreicht, wenn eine Funktionsbeeinträchtigung des Gerätes wegen unzureichender Batteriekapazität vorliegt.

Bei nicht vollständig entladenen Batterien muss Vorsorge gegen mögliche Kurzschlüsse getroffen werden. Das kann durch Isolieren der Batteriepole mit Klebestreifen erreicht werden.

2 Zusätzliche Betriebsanleitungen

Wichtig!

Die XCA Steuerung ist ein Mitglied der XCx-Steuerungsfamilie, die auf einem gemeinsamen Soft- und Hardwarekonzept basiert. Zur Ergänzung der vorliegenden Betriebsanleitung müssen daher noch folgende Betriebsanleitungen verwendet werden.

Tabelle 1: Zusätzliche Betriebsanleitungen

Bezeichnung	Artikel-Nr. bzw. Referenz
Zur Inbetriebnahme der Feldbusse	
EMV-Richtlinien deutsch	R4.322.1060.0
Inbetriebnahmehinweise für Feldbussysteme	R4.322.1600.0
Für die Programmierung der SPS und der CNC	
MULTIPROG Programmiersystem nach IEC 61131-3	MULTIPROG-Handbuch deutsch (Quickstart_MWT.pdf) im Installationspfad von MULTIPROG
CNC-Programmierung XCx und ProNumeric	R4.322.2080.0
Koppelspeicherbelegung der XCx	Online-Hilfe des Softwarepaketes zur XCx
Betriebsanleitung sercos III-I/O	R6.322.0770.0
Für die Baugruppenträger, Netzteile und Erweiterungsmodule	
Erweiterungsmodule für XCx	R4.322.2400.0

Alle Betriebsanleitungen sind als PDF-Dateien auf der Service-CDROM zur XCx verfügbar und können kostenlos von der Website *http://www.schleicher-electronic.com* geladen werden.

3 Systemübersicht

Die Steuerungen der Gruppe XCA 1100 und XCA 1200 sind weder eine SPS noch ein IPC im klassischen Sinne, sondern entsprechen in ihren grundlegenden Systemeigenschaften dem weiterführenden Konzept eines ***P***rogrammable ***A***utomation ***C***ontrollers (PAC).

Die Steuerungen XCA 1100 und XCA 1200 arbeiten mit der als VxWin bekannten Kombination aus dem Echtzeitbetriebssystem VxWorks und Windows Embedded. Hierbei übernimmt VxWorks den Echtzeitteil, d.h. die Kontrolle über die SPS-, CNC- und Motion-Control-Funktionalität, während Windows für zeitunkritische Funktionen wie Visualisierung oder Bediendialoge die gewohnte Umgebung bereitstellt:

- NC-Bediendialoge
- Visualisierung
- NC-Programmspeicher
- Diagnose
- Konfiguration
- SPS-Programmierung
- Handbuch
- Betriebsdatenerfassung

Mit Hilfe spezieller Windows-Systemtreiber und der Memory-Management-Unit (MMU) der XCA Steuerungen wird die unabhängige Arbeitsweise beider Betriebssysteme und eine saubere Trennung der Speicherbereiche realisiert. Eventuelle Instabilitäten auf der Windows-Ebene können somit keine Auswirkungen auf die Schleicher-Firmware und die SPS-/CNC-Runtime unter VxWorks haben.

In den Steuerungen XCA1100 und XCA1200 übernimmt der RTS Real-Time Hypervisor auf modernen Multi-Core Prozessoren und mit Hilfe von Intel® VT-x[1] die exklusive Zuteilung von Prozessorkernen und Speicherbereichen für den Betrieb mehrerer voneinander unabhängiger Betriebssysteme. Mit dem Hypervisor wird eine nahezu 100%-ige Trennung erreicht und ein rückwirkungsfreier Betrieb der Betriebssysteme sicherstellt.
Windows - für Visualisierung, Kommunikation und Datenspeicherung zuständig - arbeitet mit hoher Performance im virtualisierten Modus. Das für die SPS-/CNC-Steuerung entscheidende Betriebssystem VxWorks arbeitet dagegen ohne zusätzliche Latenzen und zur Gewährleistung der harten Echtzeit im „privilegierten Modus“.
Die Bootreihenfolge ist konfigurierbar, ein unabhängiges Reboot eines Betriebssystems jederzeit möglich. Die parallel laufenden Betriebssysteme kommunizieren über Shared-Memory bzw. ein Netzwerk mit virtuellen Netzwerkkarten.

[1] Intel Virtualization Technology

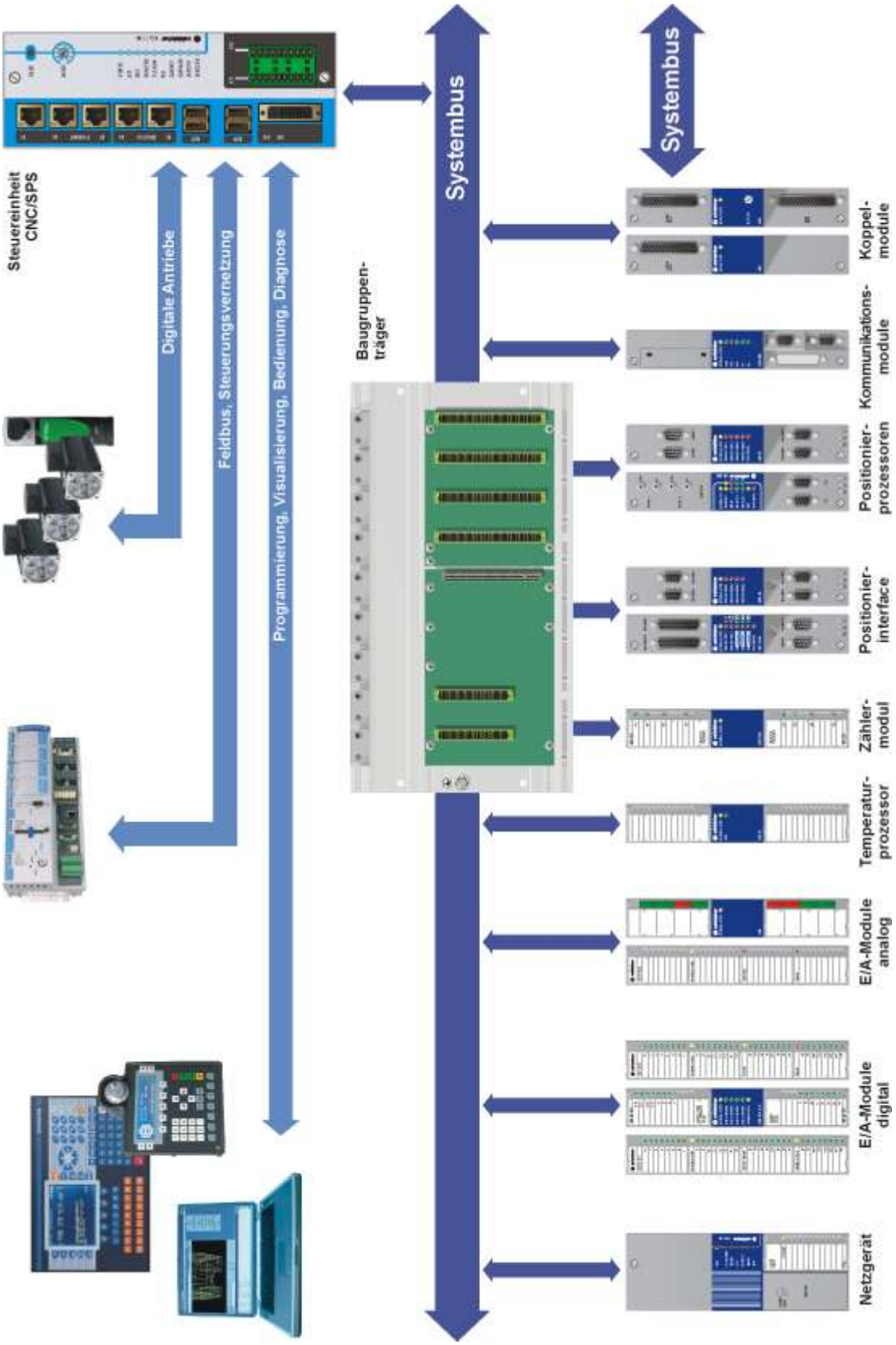


Bild 1: Systemübersicht XCA Steuerungen , Erweiterungsmodule und Peripherie

3.1 Steuerungsaufbau

Das Automatisierungssystem XCA ist modular aufgebaut, es können bis zu 256 Module auf maximal 16 Baugruppenträgern angeordnet werden.

Wichtig!

Der Einbau des Automatisierungssystemes muss in geerdeten metallischen Gehäusen (z.B. Schaltschränken) erfolgen. Es sind die in der Dokumentation "EMV-Richtlinien für den Aufbau von Automatisierungsgeräten" (Seite 10) dargelegten Vorschriften zu beachten.

Die CPU ist ein Modul im Promodul-U System-Design. Bauhöhe und –tiefe sind entsprechend ausgeführt. Das CPU-Modul (CPU und Kühlkörper) belegt insgesamt eine Breite von vier Standard-U-Modulen.

Um eine möglichst hohe Betriebssicherheit und Wartungsarmut zu erzielen, wird auf verschleißanfällige Komponenten wie Lüfter oder Festplatten verzichtet.

Zur Programm- und Datenspeicherung kommt eine Solid State Disks zur Anwendung.

Die Backplane ist mechanisch geteilt ausgeführt. Die U-Peripherie befindet sich rechts von der CPU, auf der linken Seite ist das Netzgerät angeordnet. Dieses Modul versorgt sowohl die CPU als auch den U-Bus mit den erforderlichen Betriebsspannungen.

Bild 2: Aufbau des Gesamtsystems

Netzgerät	Steuerung (Kühlkörper | CPU)	U-Erweiterungs-module

3.2 Montage

Der Steckplatz der Steuerung auf einem Baugruppenträger der XBT-Reihe (Bild 3) befindet sich zwischen dem Netzgerät (links) und den Erweiterungsmodulen (rechts). Diese Steckplatzreihenfolge ist unbedingt einzuhalten!

Bild 3: Steckplatz der Steuerung auf dem Baugruppenträger

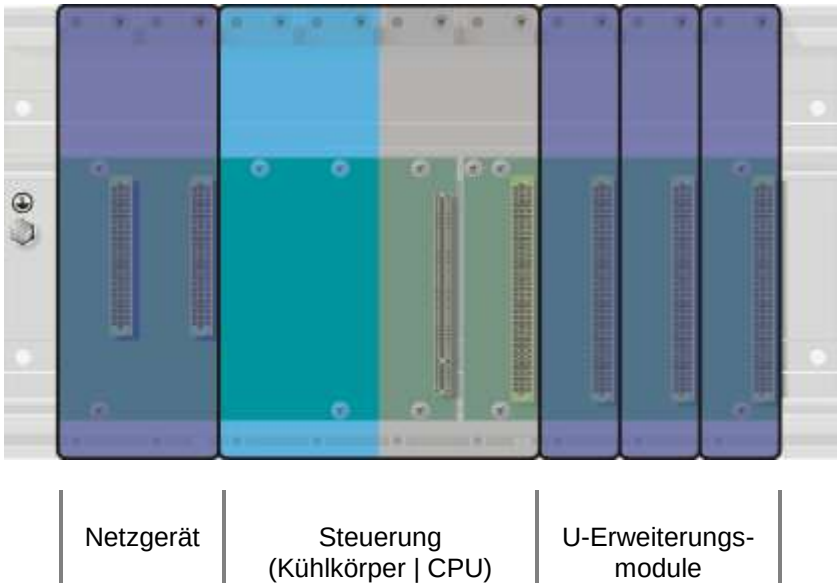
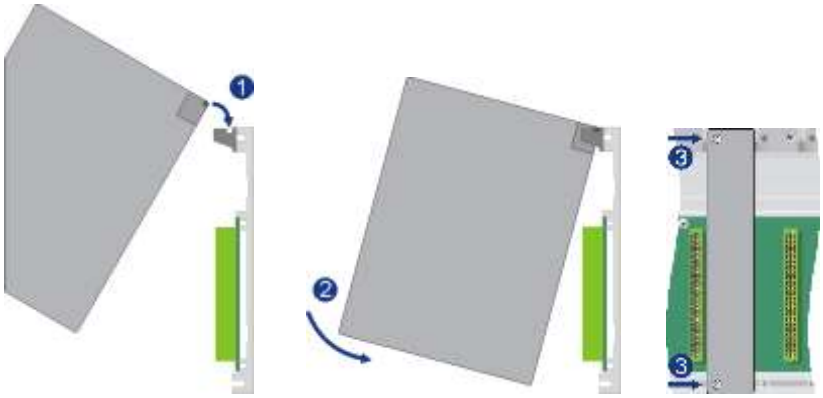


Bild 4: Montage der Module auf dem Baugruppenträger



1.
Modul mit den seitlichen Zapfen von oben in den Baugruppenträger einhängen

2.
Modul fest in die Kontaktleiste drücken

3.
Befestigungs-schrauben anziehen

Hinweis

Weitere Hinweise zum Gesamtsystem, zur Bestückung der Baugruppenträger, zur Bemessung der erforderlichen Netzgerätleistung sowie zur Auswahl der Erweiterungsmodule finden Sie in der Betriebsanleitung "Erweiterungsmodule für XCx und Promodul-U" (Seite 10)

4 Steuereinheit XCA 1100 / XCA 1200

Bild 5: XCA Steuereinheit, CPU und Kühlkörper

Die XCA Steuereinheiten sind mit einem SPS-Betriebssystem und einem leistungsfähigen CNC-Betriebssystem ausgerüstet.

Für Visualisierung, Bedienung und Programmierung steht zusätzlich ein Windows-Betriebssystem zur Verfügung.

SPS

- Betriebssystem: ProConOS
- Programmierung: MULTIPROG nach IEC 61131-3

CNC

- Programmierung: nach DIN 66025
- Maschinenspezifische Sonderfunktionen und Transformationen
- Kommunikation mit der SPS über Koppelspeicher

Windows in Abhänigkeit von der Steuerung

- Windows Embedded Standard 7
- Windows XP Embedded
- Windows 8 Embedded

Alle Steuereinheiten besitzen:

- 1 Solid State Disk
- 3 Ethernet-Schnittstellen mit integriertem Ethernet-Switch
- 4 USB 2.0 Schnittstellen
- DVI-Schnittstelle
- serielle Schnittstellen
- integrierten Webserver

Varianten:

- verschiedene Prozessor- und Speicherausstattung
- sercos III
- CANopen

(zu Einzelheiten der Varianten siehe "Varianten XCA 1100 und XCA 1200", S. 25)

4.1 Schnittstellen, Bedienelemente, Anzeigen

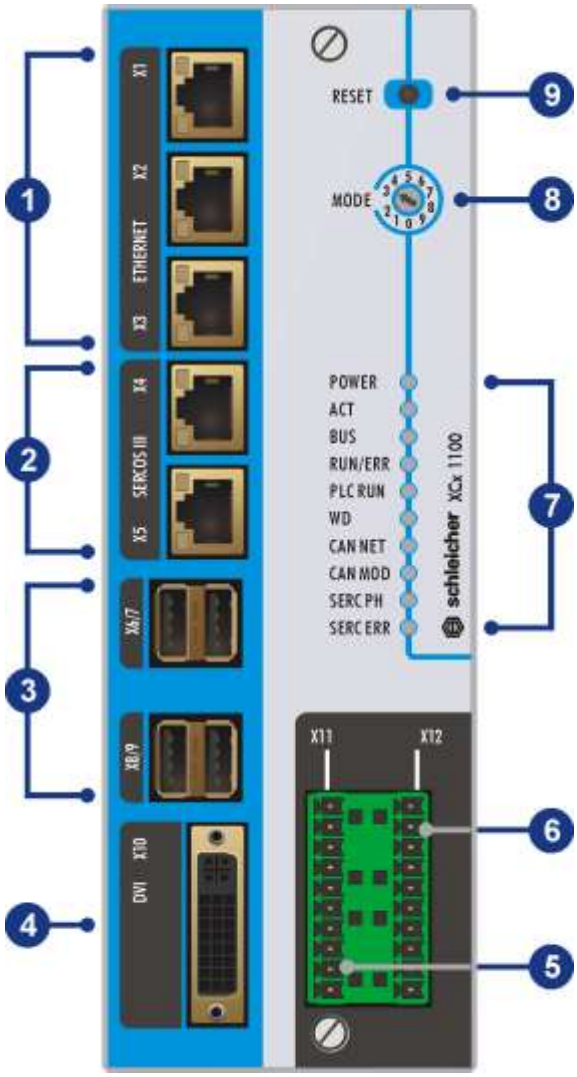


Bild 6: XCA Steuereinheit, Schnittstellen, Bedienelemente und LED-Anzeigen

1 X1, X2, X3
Ethernet-Schnittstellen, RJ 45

2 X4, X5
sercos III-Schnittstellen, RJ 45

3 X6/7, X8/9
USB-Schnittstellen

4 X10
DVI-Schnittstelle

5 X11
CAN-Schnittstellen

6 X12
RS 232 / RS 422 / RS 485
für den Anschluss von Bedien- und Anzeigegeräten

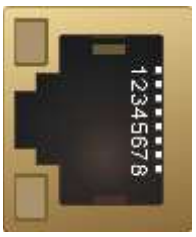
7 LED-Anzeigen

8 Betriebsartenschalter

9 Reset-Taster

4.1.1 X1, X2, X3 – Ethernet-Schnittstellen

Tabelle 2: Pinbelegung der Ethernet-Schnittstellen X1, X2, X3 (RJ 45)

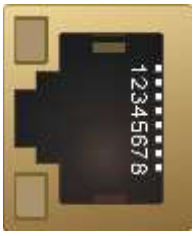


RJ 45 Buchse

Pin	Bezeichnung	Erläuterung
1	TX+	Sendedaten plus
2	TX-	Sendedaten minus
3	RX+	Empfangsdaten plus
4	nc	nicht angeschlossen
5	nc	nicht angeschlossen
6	RX-	Empfangsdaten minus
7	nc	nicht angeschlossen
8	nc	nicht angeschlossen

4.1.2 X4, X5 – sercos III-Schnittstellen

Tabelle 3: Pinbelegung der sercos III-Schnittstellen X4, X5



RJ 45 Buchse

Pin	Bezeichnung	Erläuterung
1	TX+	Sendedaten plus
2	TX-	Sendedaten minus
3	RX+	Empfangsdaten plus
4	nc	nicht angeschlossen
5	nc	nicht angeschlossen
6	RX-	Empfangsdaten minus
7	nc	nicht angeschlossen
8	nc	nicht angeschlossen

Die sercos III-Schnittstellen sind nur bei entsprechend ausgestatteten Steuereinheiten aktiv (siehe "Varianten XCA 1100 und XCA 1200", S. 25).

4.1.3 X6/7, X8/9 – USB 2.0-Schnittstellen

Tabelle 4: Pinbelegung der USB-Schnittstellen X6/7, X8/9

USB Buchse Standard A

Pin	Bezeichnung	Erläuterung
1	VCC	+5 V
2	D-	Data minus
3	D+	Data plus
4	GND	Ground

4.1.4 X10 – DVI-Schnittstelle

Die Monitor-Schnittstelle ist als DVI-I (Single-Link mit 18+5 Kontakten) ausgeführt. Es können sowohl digitale als auch (über DVI-VGA-Adapter) analoge Monitore betrieben werden.

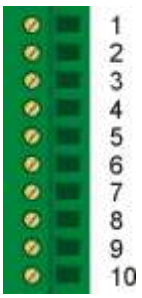
Tabelle 5: Pinbelegung der DVI-Schnittstelle X10

Pin	Bezeichnung	Erläuterung
1	TDMS Data 2-	Digital Rot minus (Link 1)
2	TDMS Data 2+	Digital Rot plus (Link 1)
3	TDMS Data 2/4 Shield	Abschirmung Daten 2,4
4	nc	nicht angeschlossen
5	nc	nicht angeschlossen
6	DDC Clock	DDC Takt
7	DDC Data	DDC Daten
8	Analog Vertical Sync	V-Sync
9	TDMS Data 1-	Digital Grün minus (Link 1)
10	TDMS Data 1+	Digital Grün plus (Link 1)
11	TDMS Data 1/3 Shield	Abschirmung Daten 1,3
12	nc	nicht angeschlossen
13	nc	nicht angeschlossen
14	+5V	
15	Ground	Masse für 5 V
16	Hotplug-Detect	
17	TDMS Data 0-	Digital Blau minus (Link 1)
18	TDMS Data 0+	Digital Blau plus (Link 1)
19	TDMS Data 0/5 Shield	Abschirmung Daten 0,5
20	nc	nicht angeschlossen
21	nc	nicht angeschlossen
22	TDMS Clock Shield	Abschirmung Takt
23	TDMS Clock+	Takt plus
24	TDMS Clock-	Takt minus
C1	Analog Red	Analog Rot
C2	Analog Green	Analog Grün
C3	Analog Blue	Analog Blau
C4	Analog Horizontal Sync	H-Sync
C5	Analog Ground	Masse

DVI-I Buchse
Single Link

4.1.5 X11 – CAN-Schnittstellen

Tabelle 6: Pinbelegung der CAN-Schnittstellen X11



Schraubblock-klemme 10-polig

Pin	Bezeich-nung	Erläuterung
1	**V+**	Stromversorgung +24DCV
2	**CAN_H**	CAN High
3	**DRAIN**	Schirmanschluss (optional)
4	**CAN_L**	CAN Low
5	**V-**	Ground 0V
6	**V+**	Stromversorgung +24DCV
7	**CAN_H**	CAN High
8	**DRAIN**	Schirmanschluss (optional)
9	**CAN_L**	CAN Low
10	**V-**	Ground 0V

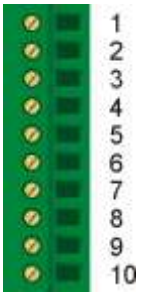
Die Pingruppen 1..5 und 6..10 sind parallel geschaltet.

Die CAN-Schnittstellen sind nur bei entsprechend ausgestatteten Steuereinheiten aktiv (siehe "Varianten XCA 1100 und XCA 1200", S. 25).

Die Schraubblockklemme ist kodiert, um ein Vertauschen der Schnittstellen X11/X12 zu verhindern.

4.1.6 X12 – RS 232 / RS 422 / RS 485-Schnittstellen

Tabelle 7: Pinbelegung der RS 232 / 422 / 485-Schnittstellen X12



Schraubblock-klemme 10-polig

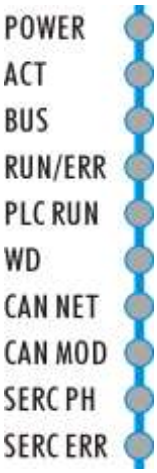
Pin	Bezeich-nung	Erläuterung
1	**SHLD**	Schirm RS 232
2	**TxD**	RS 232 Sendedaten
3	**RxD**	RS 232 Empfangsdaten
4	**M_{ext}**	Masse für RS 232
5	**M_{ext}**	Masse für RS 422 / RS 485
6	**TD-**	Sendedaten / Sende- und Empfangsdaten
7	**TD+**	Sendedaten / Sende- und Empfangsdaten
8	**RD-**	Empfangsdaten / Busabschlusswiderstände
9	**RD+**	Empfangsdaten / Busabschlusswiderstände
10	**SHLD**	Schirm RS 422 / RS 485

Die RS-Schnittstellen dienen dem Anschluss von Bedien- und Anzeigegeräten. Bei Benutzung der RS 485-Schnittstelle sind zur Aktivierung der Busabschlusswiderstände RD+ mit TD+ und RD- mit TD- zu verbinden. Die beiden Masse-Pins für RS 232 und RS 485 haben das gleiche Potenzial.

Die Schraubblockklemme ist kodiert, um ein Vertauschen der Schnittstellen X11/X12 zu verhindern.

4.1.7 LED-Anzeigen Steuerung

Tabelle 8: LED-Anzeigen auf der Steuerung



Bezeichnung	Farbe	Zustand	Bedeutung
POWER			**Power**
		aus	Gerät ausgeschaltet
	grün	ein	Gerät eingeschaltet
ACT			**IDE (CF) / SATA (SSD) Aktivität**
		aus	kein Zugriff
	grün	blinkend	Zugriff erfolgt
BUS			**Buszugriff**
		aus	kein Buszugriff; bei SPS-Stop oder (Echtzeit-)Betriebssystem nicht aktiv
	grün	ein	Buszugriff in Ordnung
	rot	blinkend	Buszugriffsfehler / Konfigurationsfehler
RUN/ERR			**CPU-Status**
		aus	CPU defekt
	grün	ein	CPU bootet
	grün	ein	CPU läuft, Betriebsspannung in Ordnung, kein Fehler
	rot	blinkend	fataler Fehler: CPU kann nicht booten
PLC RUN			**SPS-Status**
		aus	SPS Stop
	grün	ein	SPS läuft
	gelb	blinkend	SPS läuft, aber Ausgänge sind abgeschaltet (Betriebsbereit-Relais abgefallen)
WD			**Watchdog**
		aus	Watchdog hat nicht angesprochen
	rot	ein	schwerwiegender Fehler oder (Echtzeit-)Betriebssystem nicht aktiv
CAN NET			**CAN Netzwerkwerkstatus**
		aus	CAN State Prepared
	grün	ein	CAN State Operational
		blinkend	CAN State Pre-Operational
	rot	ein	Bus Off
		blinkend	CAN-Fehler
CAN MOD			**CAN Modulstatus**
	grün	ein	CAN-Stack initialisiert
		blinkend	ungültige CAN-Konfiguration
	rot	ein	Steuereinheit nicht bereit oder schwerer Fehler
		blinkend	Fehler in der Steuerung

(Fortsetzung)

Tabelle 8: LED-Anzeigen auf der Steuerung (Fortsetzung)

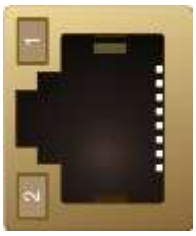
Bezeichnung	Farbe	Zustand	Bedeutung
SERC PH			**sercos Phasen**
	rot	ein	sercos Phase 0
	rot	blinkend	sercos Phase 1
	gelb	blinkend	sercos Phase 2
	grün	blinkend	sercos Phase 3
		ein	sercos Phase 4
SERC ERR			**sercos Fehler**
	rot	aus	kein Fehler
		ein	Kommunikationsfehler
		blinkend	Antriebsfehler

Fehlermeldungen werden im Active-Error-Buffer und im Error-Logbook gespeichert, sie sind mit Fehlernummern und zusätzlichen Angaben gekennzeichnet.

Active-Error-Buffer und Log-Book sind im Schleicher-Dialog auf jeder Bedienebene über die Tastenkombination <Ctrl+?> aufrufbar.

4.2 LED-Anzeigen Ethernet und sercos III

Tabelle 9: LED-Anzeigen an den Ethernet- und sercos-III-Buchsen (5x RJ45)



Bezeichnung	Farbe	Zustand	Bedeutung
ETH (3x)			**Duplex**
LED 1		aus	keine Netzwerkverbindung oder 10 Mbits/s
	grün	ein	100 MBit/s Full Duplex Betrieb
LED 2			**Link / Activity / Speed**
		aus	keine Netzwerkverbindung
	gelb	blinkend	100 MBit/s Verbindung aktiv
	grün	blinkend	10 MBit/s Verbindung aktiv
sercos III	**(2x)**		**Link**
LED 1		aus	keine Netzwerkverbindung
	grün	ein	Netzwerkverbindung hergestellt
LED 2			**Activity**
		aus	keine Aktivität
	gelb	blinkend	100 MBit/s Verbindung aktiv

4.2.1 Betriebsartenschalter

Der Betriebsartenschalter verfügt über zehn Stellungen, mit denen das Hochlaufverhalten der Steuerung bestimmt wird.

Tabelle 10: Betriebsartenschalter

Stellung / Bezeichnung	Bedeutung
0	Grundinitialisierung / Diagnose (Start des Echtzeitbetriebs-systems im abgesicherten Modus und Rücksetzen des remanenten Datenspeichers, S. 79)
1 / Prog	Betriebsart Programmierung (SPS-Stopp)
2 / Warm (auch 4..9)	Warmstart der SPS nach IEC 61131-3 (Defaultstellung)
3 / Cold	Kaltstart der SPS nach IEC 61131-3 (Reinitialisierung der Retainvariablen)

Die aktuelle Einstellung des Betriebsartenschalters kann im SPS-Programm abgefragt werden. In den mit MULTIPROG mitgelieferten Templates für die XCA-Steuereinheiten ist die Variable bereits angelegt (Arbeitsblatt *Global_Variables, PLC_COMMON:* `cmpSwrdPlcRd_lXModeSwitch`).

4.2.2 Reset-Taster

Der Reset-Taster ermöglicht das Abschalten und Rücksetzen des CPU-Moduls:

- kurzer Tastendruck = Reset
- langer Tastendruck = Abschalten

Tabelle 11: Reset-Taster

Wichtig!

Wird RESET bei Zugriffen auf die SSD ausgelöst (LED *ACT* blinkt grün), kann Datenverlust auftreten.

4.3 Technische Daten Steuereinheit XCA

Elektrische Daten		
Versorgungsspannung intern		DC 12 V, DC 5 V, DC 3,3 V
Leistungsaufnahme intern		< 40 W
Galvanische Trennung (zur internen Elektronik)	X1, X2, X3 Ethernet	Ja
	X4, X5 sercos III	Ja
	X6/7, X8/9 USB	Nein
	X10 DVI	Nein
	X11 CAN	Ja
	X12 (RS 422)	Ja
	X12 (RS 232)	Ja

Schnittstellen		
Ethernet	RJ 45	Programmier-, Diagnose und Bediengeräteschnittstelle
sercos III	RJ 45	sercos III Antriebsschnittstelle (Ethernet)
USB	Standard A	USB-Schnittstelle (z.B. Maus, Tastatur, USB-Speicherstick)
DVI	DVI-I Single Link	Monitor-Schnittstelle (DVI-I)
CAN	10-pol. Steckblockklemme	CANopen Feldbusschnittstelle
RS 232	10-pol. Steckblockklemme	für stationären Anschluss serieller Geräte
RS 422	10-pol. Steckblockklemme	serielle Bediengeräteschnittstelle

Hardware und Speicher		
Prozessor		Leistungsvarianten Verschiedene Prozessoren, bis Core i7, sind möglich.
Speicher-ausstattung	SDRAM	512 MB bis 4 GB
	SRAM (gepuffert)	1 MB
	Solid State Drive	32 GB / 64 GB
Real-Time Clock		Batteriegepuffert mit Kalender und Schaltjahr, Auflösung: 1s
Pufferung		Supercap min. 3 Std, wiederaufladbare Batterie nach mindestens 4Std Ladezeit, min.3 Mon.

CNC/SPS-Eigenschaften		
SPS-Bearbeitungs-zeiten je 1000 Anweisungen	Bit	0,064 ms
	Byte / Word / DWord	0,033 ms
	Integer (Add / Mul)	0,038 ms
	Real (Add)	0,064 ms
SPS-Signallaufzeit (Input to Output)		< 2 ms (bei Taskperiode = 1 ms)
Funktionsbausteine		Firmwarefunktionen und Funktionsbausteine in beliebiger Anzahl
Anzahl NC-Achsen / Teilsysteme		64 / 32
CNC-Interpolationstakt, ab		1 ms
Blockzykluszeit, ab		1 ms
Betriebs-system	Steuerung	VxWorks, Multitask-Betriebssystem (zeit- und prioritätsgesteuert)
	SPS-Runtime	ProConOS
	PC	Windows Embedded
Projektierung		MULTIPROG nach IEC 61131-3
Anzahl der Anwendertasks		18
Taskzykluszeiten		programmierbar ≥ 1 ms (ganzzahlig)
Echtzeit-Speicher (einstellbar)	Betriebssystem (Daten / Programme)	32768 kB
SPS-Speicher	Programme	4096 kB
	Merker remanent	256 kB
	Merker nicht remanent	2048 kB
Speicherverwaltung		dynamisch
Zeiten und Zähler		beliebig viele programmierbar von 1 ms ... 290 h (Anzahl nur durch Speicherauslastung begrenzt)

Abmessungen / Gewicht	
Maße (B x H x T)	142 mm x 200 mm x 150 mm
Teilungsbreite	4
Gewicht	2500 g

Zusätzlich gelten die Angaben im Kapitel "Technische Daten aller Module", Seite 130.

4.4 Varianten XCA 1100 und XCA 1200

Varianten für XCA 1100 und XCA 1200	
XCA 1100	Celeron M, 1,5 GHz, 512 MB, 4 GB Compact Flash
XCA 1110	Celeron M, 1,5 GHz, 1 GB DDR2-SODIMM, 32 GB SSD
XCA 1120	Core 2 Duo, 1,2 GHz, 1 GB DDR2-SODIMM, 32 GB SSD
XCA 1125	Core 2 Duo, 1,8 GHz, 1 GB DDR2-SODIMM, 32 GB SSD
XCA 1200	Core 2 Duo, 1,8 GHz, 2 GB, DDR2-SODIMM, 64 GB SSD, Hypervisor
Basisfunktionen	
	IPC Steuereinheit für 64 interpolierende Achsen
	PC-basierte Hochleistungssteuereinheit
	CNC-Betriebssystem mit synchr. SPS
	sercos III I/O
	Für mehr als 4 Achsen wird eine Software-Option SCR xx benötigt
	Windows Embedded
	Ethernet Switch
	DVI
	USB
Optionale Funktionen	
C	CANopen
S	sercos III für Antriebe
E	Exportversion, IPC Steuereinheit für max. 4 interpolierende Achsen
Software-Option	
SRC8	8-Achsen sercos III
SRC16	16-Achsen sercos III
SRC24	24-Achsen sercos III
SRC32	32-Achsen sercos III

5 Erweiterungsmodule XCx

Für die XCA Steuereinheiten stehen eine Vielzahl an Baugruppenträgern, Netzgeräten und Erweiterungsmodulen zur Verfügung. Diese Module werden in einer eigenen Betriebsanleitung beschrieben (Seite 10).

Modul	Artikel-Nr.	Bemerkung
Baugruppenträger		
XBT 0 / 1100	R4.507.0040.0	Basisbaugruppenträger, NT, CPU
XBT 3 / 1100	R4.507.0010.0	Basisbaugruppenträger, NT, CPU und 3 Steckplätze
XBT 4 / 1100	R4.507.0030.0	Basisbaugruppenträger, NT, CPU und 4 Steckplätze
XBT 7 / 1100	R4.507.0020.0	Basisbaugruppenträger, NT, CPU und 7 Steckplätze
XBT 11 / 1100	R4.507.0050.0	Basisbaugruppenträger, NT, CPU und 11 Steckplätze
UBT 4 x*	R4.311.0010.0	Basis-/Erweiterungsbaugruppenträger, 4 Steckplätze
UBT 8 x*	R4.311.0020.0	Basis-/Erweiterungsbaugruppenträger, 8 Steckplätze
UBT 12 x*	R4.311.0030.0	Basis-/Erweiterungsbaugruppenträger, 12 Steckpl.
UBT 16 x*	R4.311.0040.0	Basis-/Erweiterungsbaugruppenträger, 16 Steckpl.
Koppelmodule		
UKZ	R4.318.0030.0	Koppelmodul für Basisbaugruppenträger
UKE	R4.318.0040.B	Koppelmodul für Erweiterungsbaugruppenträger
Netzgeräte		
XNG 24	R4.507.0100.0	Netzgerät 24 V, Breite 2 Einheiten
UNG 230A x**	R4.312.0030.F	Netzgerät 230 V, Breite 2 Einheiten
UNG 115A x**	R4.312.0040.F	Netzgerät 115 V, Breite 2 Einheiten
UNG 24 x**	R4.312.0020.B	Netzgerät 24 V, Breite 1 Einheit
Digitale E/A-Module		
UBE 32 0,1I	R4.314.0100.E	32 Eingänge, 4 Eingangsverzögerung, 0,1 ms Eingangsverzögerung.
UBE 32 1D	R4.314.0120.E	32 Eingänge, 1 ms Eingangsverzögerung
UBE 32 10D	R4.314.0090.E	32 Eingänge, 10 ms Eingangsverzögerung
UBA 32/2A	R4.314.0080.D	32 Halbleiterausgänge DC 24V / 2A
UBK 16E 1D/16A	R4.314.0130.E	16 Eingänge, 1 ms Eingangsverzögerung. / 16 Ausgänge
UBK 16E 10D/16A	R4.314.0110.E	16 Eingänge, 10 ms Eingangsverzögerung. / 16 Ausgänge
XBE 32 1D	R4.314.0140.0	32 Eingänge, 1 ms Eingangsverzögerung
XBE 32 10D	R4.314.0180.0	32 Eingänge, 10 ms Eingangsverzögerung
XBE 32 0,1I	R4.314.0170.0	32 Eingänge, 4 Interrupts, 0,1 ms Eingangsverzögerung.
XBA 32/1A	R4.314.0150.0	32 Halbleiterausgänge DC 24V / 1A
XBK 16E 1D/16A	R4.314.0160.0	16 Eingänge, 1 ms Eingangsverzögerung / 16 Ausgänge
XBK 16E 10D/16A	R4.314.0190.0	16 Eingänge, 10 ms Eingangsverzögerung / 16 Ausgänge
Zählermodule		
UZB 2VR	R4.315.0010.B	2 Zähler, 24 V Eingangsspannung
UZB 2VR/5V	R4.315.0040.B	2 Zähler, 5 V Eingangsspannung

Modul	Artikel-Nr.	Bemerkung
Analog- und Temperaturmodule		
UAK12E/4A	R4.315.0230.0	12 Eingänge 0..10 V, 4 Ausgänge ±10 V
USA 8/1	R4.315.0090.F	Analogprozessor, 8 Slots für USA-Module
USA E1/1	R4.315.0100.0	1 Spannungseingang
USA E1/2.1	R4.315.0120.0	1 Stromeingang
USA E1/6	R4.315.0140.0	1 Widerstandstemperaturmessung Pt100
USA E1/7	R4.315.0150.0	1 Thermoelementeingang Fe-CuNi
USA A1/1	R4.315.0110.B	1 Spannungsausgang
USA A1/2	R4.315.0130.0	1 Stromausgang
UST 2	R4.315.0170.0	Temperaturmodul, 8 Eingänge
UST 21	R4.315.0180.0	Temperaturmodul, 8 Eingänge, adaptive Regelung
Positioniermodule		
USP 200S	R4.315.0300.0	sercos-Master, 1 Ring, 8 Achsen, Kinematikfunkt.
USP 400S	R4.315.0330.0	sercos-Master, 2 Ringe, 16 Achsen
USP 2I	R4.315.0020.0	Positionierprozessor, 2 Achsen, Inkremental-Encoder
USP 2A	R4.315.0030.0	Positionierprozessor, 2 Achsen, Absolut-Encoder SSI
UPI 2 DIA	R4.318.0180.B	Positionierinterface, 2 Achsen
UPI 3 DIA	R4.318.0160.B	Positionierinterface, 3 Achsen
UPM 3I	R4.315.0080.B	Positionserfassung, 3 Kanäle, Inkremental-Encoder
UPM 4A	R4.315.0060.C	Positionserfassung, 4 Kanäle, Absolut-Encoder
UPM 4U	R4.315.0310.C	Positionserfassung, 4 Kanäle, Ultraschallgeber
Multifunktionsmodule		
XSF 05	R4.315.0340.0	8 I/O 24 V DC, 14 I/O 5 V DC
XSF 24	R4.315.0350.0	22 I/O 24 V DC
XSL 05	R4.315.0360.0	Für Laseransteuerung, 8 I/O 24 V DC, 14 I/O 5 V DC
XSL 24	R4.315.0370.0	Für Laseransteuerung, 22 I/O 24 V DC
Kommunikationsmodule		
USK DIM	R4.318.0170.0	Interbus-S-Master
USK DPM	R4.318.0370.0	PROFIBUS-DP-Master
USK DPS	R4.318.0360.0	PROFIBUS-DP-Slave
Zubehör		
UBT LA	R4.318.0120.0	Leerort-Abdeckungen für UBT
UKK 24	R4.318.0020.0	Kabel UKZ ↔ UKE, ohne Spannungsversorgung
UKK 24V	R4.318.0060.0	Kabel UKZ ↔ UKE, mit Spannungsversorgung
UNB 115/230	R4.318.0050.0	Pufferbatterie für UNG 230A/115A
UNB 24	R4.318.0130.0	Pufferbatterie für UNG 24
UST	R4.315.0160.F	Temperaturregler (Ersatzteil für UST 2 / UST 21)

* x* = Verwendung nur als Erweiterungsbaugruppenträger zu einem Basisbaugruppenträger XBT
x** = Verwendung nur auf Erweiterungsbaugruppenträgern

Tabelle 12: Liste der verfügbaren Baugruppenträger, Netzteile und Erweiterungsmodule für XCx und Promodul-U

6 Inbetriebnahme

Die Inbetriebnahmeschritte müssen genau befolgt und die Rahmenbedingungen (wie z. B. die I/O-Konfiguration) eingehalten werden.

Hinweis

Die in den folgenden Kapiteln gezeigten Screenshots zu Software-Installation und Inbetriebnahme sind beispielhaft. Versionsnummern von Software oder Gerätebezeichnungen können vom aktuellen Stand abweichen.

6.1 Installation von MULTIPROG, OPC-Server und AddOns

Wichtig!

Die gesamte Programmiersoftware besteht aus den Software-Komponenten MULTIPROG, dem OPC-Server, AddOns für MULTIPROG und dem Schleicher-Dialog.

Alle Software-Komponenten müssen vor der weiteren Inbetriebnahme einzeln, in dieser Reihenfolge nacheinander installiert werden.

Die Programmiersoftware kann sowohl auf einem externen PC als auch auf dem PC der Steuerung installiert werden. Nachfolgend wird die Installation auf einem externen PC beschrieben.

Wenn vom Kunden gewünscht wird die Steuerung mit zwei CD-ROM ausgeliefert:

Tabelle 13: Inhalt der CD-ROM

Name	Inhalt
MULTIPROG	• Programmiersoftware MULTIPROG • OPC-Server
Service Pack	• Steuerungssoftware für alle Schleicher-Steuerungen • AddOns • Schleicher-Dialog • Weitere Hilfsmittel wie Dokumentation und Service-Informationen

6.1.1 Systemvoraussetzungen

Für Installation und Betrieb der Software auf einem externen PC sind folgende Systemvoraussetzungen einzuhalten:

Tabelle 14: Systemvoraussetzungen

Windows-PC	Pentium 4, 2 GHz [1]
Arbeitsspeicher	512 MB [1]
Festplatte	250 MB freier Speicherplatz [1]
Monitor	1024 x 768 (True Color) [1]
Kommunikation	TCP/IP oder RS232
Maus	
PC-Betriebssystem	Microsoft Windows XP mit SP3 Microsoft Windows Vista mit SP2 Microsoft Windows 7 (32 oder 64 Bit) Microsoft Windows 8 Microsoft NET Framework 3.5 [2] Microsoft Visual C++ 2005 [3]Redistributables und Microsoft Visual C++ 2008 [3]Redistributables.

1 Mindestvoraussetzung, bessere Ausstattung zum komfortablen Arbeiten empfohlen.

2 Microsoft .NET Framework 3.5 wird nicht mitgeliefert.

3 Microsoft Visual C++ 2005 Redistributables und Microsoft Visual C++ 2008 Redistributables werden mitgeliefert.

6.1.2 MULTIPROG installieren

Die CD MULTIPROG in das Laufwerk des PC einlegen. Über die AutoRun-Funktion der CD wird der Internet Explorer gestartet. Dann *MULTIPROG* (hier Version 5.x) auswählen und die Installation starten (Bild 7).

Bild 7: Installation von MULTIPROG

Hinweis

Ist eine Version von MULTIPROG kleiner als 5 bereits installiert, darf die installierte Version nicht überschrieben werden, wenn weiter mit den alten Projekten gearbeitet werden soll.

MULTIPROG muss dann in einem neuen Pfad installiert werden. Alle anderen Installations-Einstellungen können unverändert bleiben.

Am Ende der Installation erfolgt eine Aufforderung zum Neustart des Computers. Falls Sie anschließend auch den ProConOS OPC-Server installieren wollen, ist ein Neustart des Computers jetzt noch nicht erforderlich.

6.1.3 OPC-Server installieren

Zur Installation des OPC-Servers im Internet Explorer den *ProConOS OPC-Server* auswählen und die Installation starten (Bild 7).

Der OPC-Server soll im MULTIPROG-Pfad installiert werden. Alle anderen Installations-Einstellungen können unverändert bleiben. Nach der Installation ist ein Neustart des PCs erforderlich.

6.1.4 AddOns installieren

Als nächster Schritt müssen die AddOns für MULTIPROG installiert werden.

Legen Sie dazu die CD *Service Pack* ein. Über die AutoRun-Funktion der CD wird der Internet Explorer gestartet. Dann unter der Rubrik für die vorhandene Steuerung *AddOns für MULTIPROG* auswählen und die Installation starten (Bild 8).

Bild 8:
Installation der AddOns

schleicher
Schleicher Service Pack 13.23
Willkommen zur Schleicher Service Pack CD, Version 13.23 (Stand 03. Juni 2013). Auf dieser CD finden Sie die aktuellen Versionen der Schleicher Add-Ons für MULTIPROG®, des Schleicher Dialogs und des ProConOS® OPC-Servers für die Steuerungsfamilien XCx, XCx-micro und ProNumeric / ProSyCon.
Wählen Sie einen der folgenden Links, um die entsprechende Software zu installieren.
• Schleicher Add-Ons für MULTIPROG®
Schleicher Add-Ons für MULTIPROG®, Version 13.23(0)
• Schleicher Dialog
Schleicher Dialog, Version 13.09(0)
• OPC-Server
ProConOS® OPC-Server 2.1, Build 202
ProConOS® OPC-Server 2.0, Build 42
ProConOS® OPC-Server 1.12, Build 28
Beachten Sie bitte diese Hinweise zum ProConOS® OPC-Server.

Ein Neustart des PCs nach der Installation ist nicht erforderlich.

6.1.5 Schleicher-Dialog installieren

Anschließend wird die Bedienoberfläche Schleicher-Dialog installiert. Dazu im Internet Explorer unter der Rubrik für die vorhandene Steuerung *Dialog* auswählen und starten (Bild 9).

Bild 9:
Installation des Schleicher-Dialogs

Schleicher Service Pack 13.23
Willkommen zur Schleicher Service Pack CD, Version 13.23 (Stand 03. Juni 2013). Auf dieser CD finden Sie die aktuellen Versionen der Schleicher Add-Ons für MULTIPROG®, des Schleicher Dialogs und des ProConOS® OPC-Servers für die Steuerungsfamilien XCx, XCx-micro und ProNumeric / ProSyCon.
Wählen Sie einen der folgenden Links, um die entsprechende Software zu installieren.
• Schleicher Add-Ons für MULTIPROG®
Schleicher Add-Ons für MULTIPROG®, Version 13.23/0
• Schleicher Dialog
Schleicher Dialog, Version 13.09/0
• OPC-Server
ProConOS® OPC-Server 2.1, Build 202
ProConOS® OPC-Server 2.0, Build 42
ProConOS® OPC-Server 1.12, Build 28
Beachten Sie bitte diese Hinweise zum ProConOS® OPC-Server.

6.2 Inbetriebnahme der Netzwerk-Schnittstelle

Wichtig!

Die netzwerkspezifischen Besonderheiten und die Vorgehensweise sind zuerst mit dem Netzwerkadministrator für das jeweilige Hausnetz zu klären.

Alle in den folgenden Installationshinweisen angeführten bzw. vorgegebenen Bezeichner und Adressen sind beispielhaft und müssen gegebenenfalls an Ihre lokalen Verhältnisse angepasst werden.

Alle Beispiele dieser Anleitung sind mit Windows XP erstellt. Die Vorgehensweise in anderen Betriebssystemen kann von der hier beschriebenen abweichen.

Die hier gegebenen Hinweise sind in jedem Falle unverbindlich!

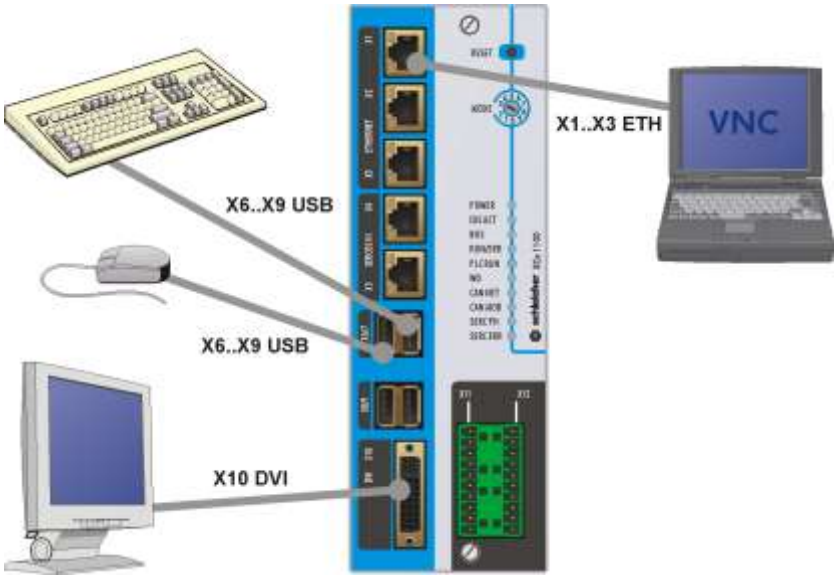
6.2.1 Vorbereitung

Zur Vorbereitung der Inbetriebnahme der Netzwerk-Schnittstelle auf der Steuerung schließen Sie folgende Geräte an (Bild 10, links):

- einen digitalen Monitor direkt (bzw. einen analogen Monitor über einen VGA-DVI-Adapter) an die DVI-Schnittstelle X10;
- eine Tastatur an eine der USB-Schnittstellen X6..X9;
- eine Maus an eine der USB-Schnittstellen X6..X9.

Die Stellung des Betriebsartenschalters ist beliebig. Starten Sie die Steuerung durch Anlegen der Betriebsspannung an das Netzteil.

Bild 10: Anschluss der Ein-/Ausgabegeräte an die XCA Steuerung



Alternativ zur direkten Eingabe ist die Inbetriebnahme mit der Fernsteuerungssoftware VNC (Seite 129) über eine Ethernet-Verbindung möglich (Bild 10, rechts). Für eine erste Verbindung mit VNC sind die nachfolgend beschriebenen IP-Adressen im PC zu verwenden.

6.2.2 Kommunikationsschema

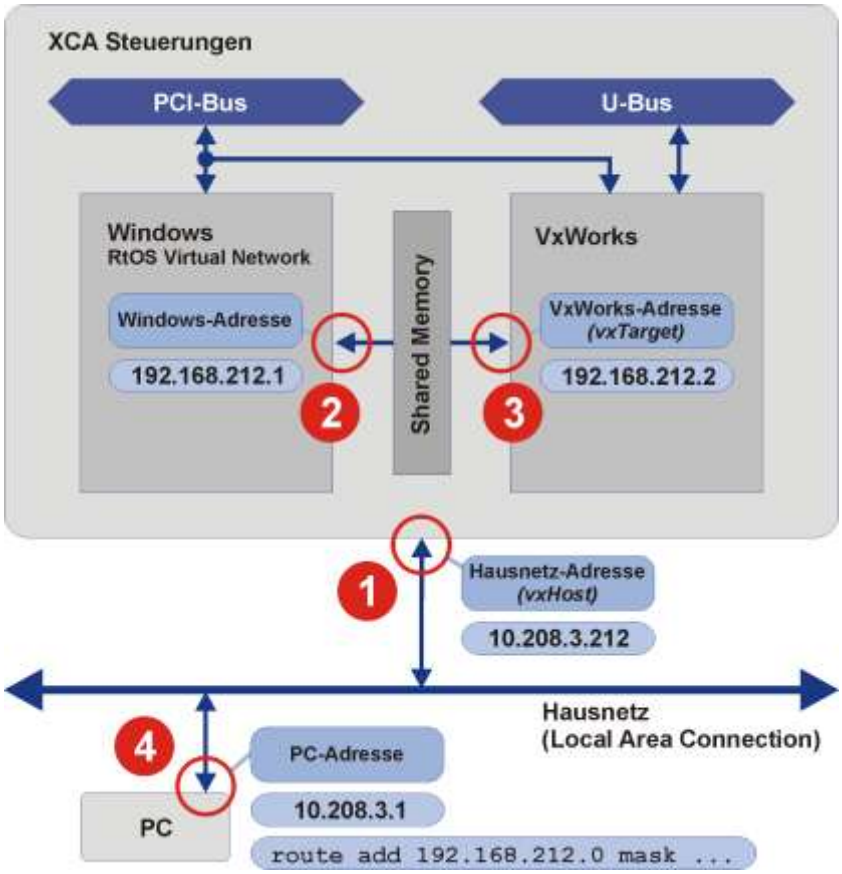


Bild 11: Inbetriebnahme der Netzwerkschnittstellen in vier Schritten

Die XCA Steuerung enthält das Echtzeitbetriebssystem VxWorks sowie einen Windows-Teil. Beide kommunizieren über einen gemeinsamen Speicher (Shared Memory) miteinander.
Über den U-Bus werden die Erweiterungsmodule und die erforderlichen Versorgungsspannungen (Netzteil) angebunden, der PCIexpress-Bus dient für künftige schnelle Erweiterungsmodule.

Im folgenden Kapitel wird die Inbetriebnahme der Netzwerkschnittstellen in vier Schritten beschrieben (Bild 11):

(1) Vergabe einer **Hausnetz-Adresse** zur Anbindung der Steuerung an ein vorhandenes Hausnetz; die angegebene IP-Adresse ist beispielhaft, der Aliasname (vxHost) ist herstellerseitig vorgegeben.

(2) Ändern der **Windows-Adresse**; nach Möglichkeit sollte die vorgegebene Adresse (192.168.212.1) beibehalten werden.

(3) Ändern der **VxWorks-Adresse**; nach Möglichkeit sollte die vorgegebene Adresse (192.168.212.2) beibehalten werden; der Aliasname (vxTarget) ist herstellerseitig vorgegeben.

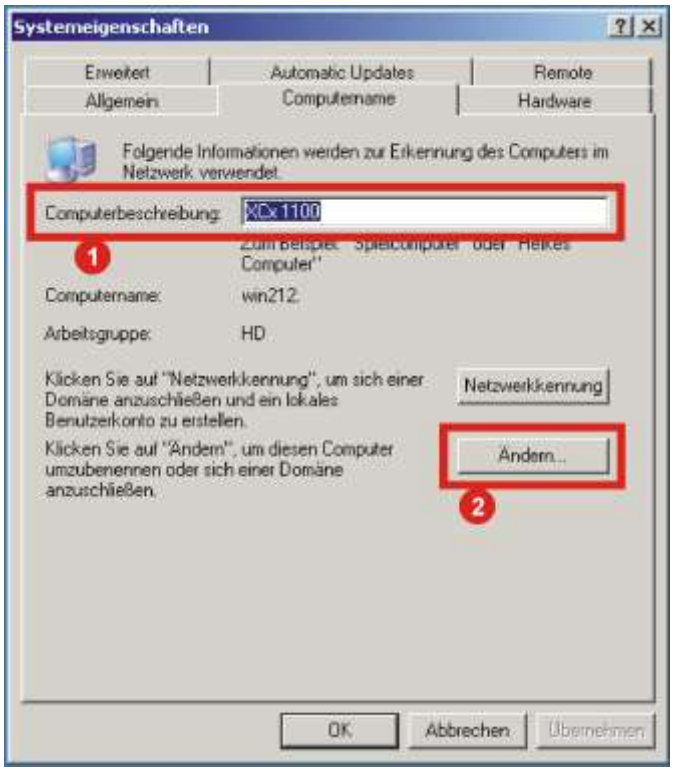
(4) Einrichten einer **PC-Adresse** zur Kommunikation mit der Steuerung über das Hausnetz (Programmierung mit MULTIPROG).

Zur vereinfachten Anpassung dieser komplexen Einstellungen an das lokale Netzwerk dient der X-Manager, siehe Schleicher X-Manager Seite 123. Die manuelle Änderung einiger ausgewählter Parameter wird in den nachfolgenden Kapiteln beschrieben.

6.2.3 Computername für XCx vergeben

Im Dialogfenster *Systemeigenschaften* (unter *Start / Systemsteuerung / System / Computername)* geben Sie als *Computerbeschreibung* z.B. "XCx 1100" ein (1) (Bild 12).

Bild 12: Dialogfenster "Systemeigenschaften", Eingabe der Computerbeschreibung



Über den Button *Ändern* (2) kommen Sie zum Dialogfenster *Computernamen ändern*, wo Sie einen beliebigen Computernamen und eine Arbeitsgruppe eintragen können (Bild 13). Der vorgegebene Name "win212" wird in den folgenden Beispielen verwendet; der Übersichtlichkeit halber sollten Sie ihn beibehalten. Fragen Sie dazu gegebenenfalls Ihren Netzwerkadministrator.

Bild 13: Dialogfenster "Computernamen ändern"

6.2.4 XCx-TCP/IP-Einstellungen anpassen

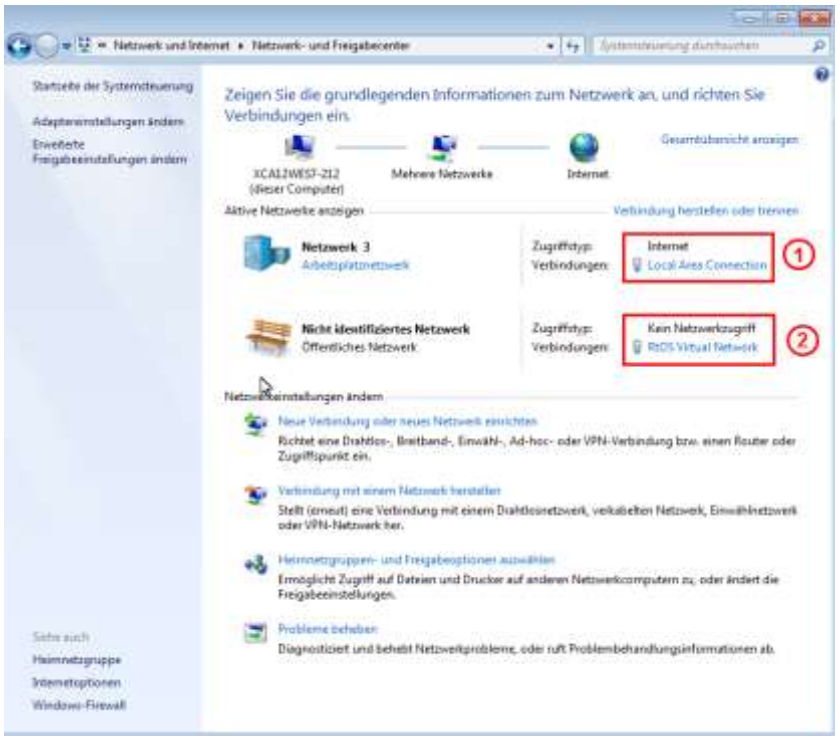
Wichtig!

Vor dem Anpassen der IP-Adressen die Vorgehensweise zuerst mit dem Netzwerkadministrator für das jeweilige Hausnetz klären.

Die hier gegebenen Hinweise sind in jedem Falle unverbindlich.

Um die nötigen TCP/IP-Einstellungen der Netzwerkkarte vorzunehmen, öffnen Sie über *Start / Systemsteuerung* die *Netzwerkverbindungen* (Bild 14).

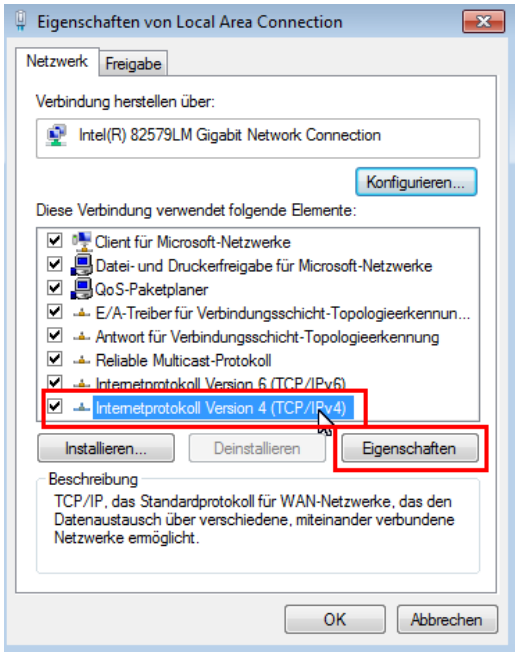
Bild 14:
Auswahl der Netzwerk-verbindungen



IP-Adressen für Local Area Connection (Schritt 1)

Ein Doppelklick auf *Local Area Connection* (1) öffnet das zugehörige Eigenschaftsfenster. Dort wählen Sie *Internet Protokoll (TCP/IP)* und klicken auf den Button *Eigenschaften* (Bild 15).

Bild 15: Eigenschaften von Local Area Connection

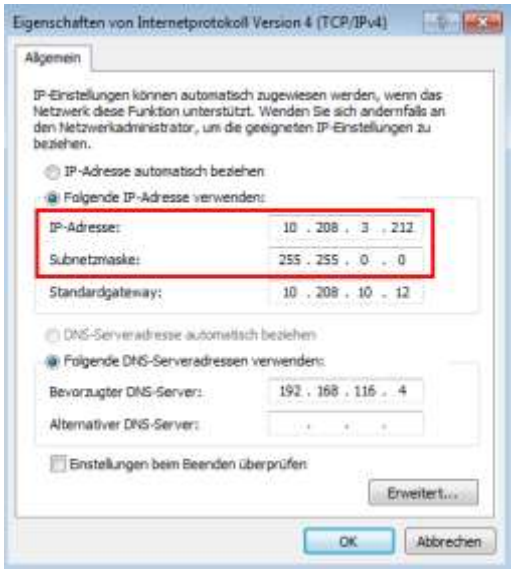


Im nächsten Fenster *Eigenschaften von Internet Protocol (TCP/IP)* übernehmen Sie die vorgegebenen Werte (Bild 16):

- IP-Adresse: 10.208.3.212 (default)
- Subnetzmaske: 255.255.0.0 (default)

Auch die übrigen Einträge können unverändert übernommen werden.

Bild 16: Eintragen von IP-Adresse und Subnetzmaske für Local Area Connection


Eigenschaften von Internetprotokoll Version 4 (TCP/IPv4)
Allgemein
IP-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IP-Einstellungen zu beziehen.
IP-Adresse automatisch beziehen
Folgende IP-Adresse verwenden:
IP-Adresse: 10 . 208 . 3 . 212
Subnetzmaske: 255 . 255 . 0 . 0
Standardgateway: 10 . 208 . 10 . 12
DNS-Serveradresse automatisch beziehen
Folgende DNS-Serveradressen verwenden:
Bevorzugter DNS-Server: 192 . 168 . 116 . 4
Alternativer DNS-Server:
Einstellungen beim Beenden überprüfen
Erweitert...
OK
Abbrechen

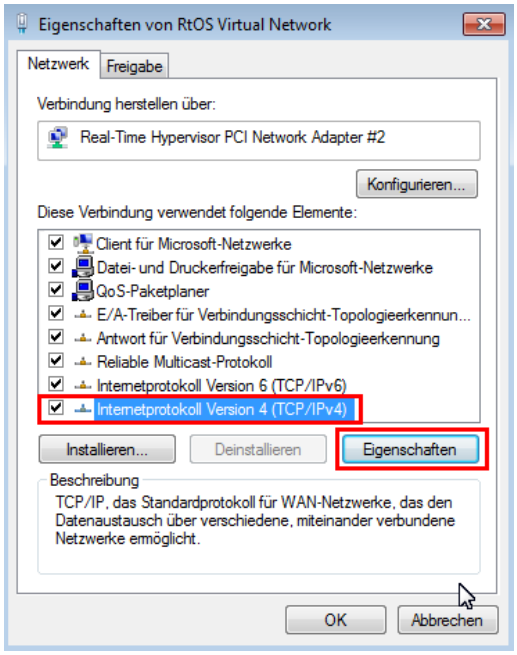
Wichtig!

Für die Installation mit Anbindung an ein vorhandenes Hausnetz ist vor dem Anpassen der IP-Adressen die Vorgehensweise mit dem Netzwerkadministrator für das jeweilige Hausnetz zu klären.

IP-Adressen für RtOS Virtual Network (Schritt 2)

Um die IP-Einstellungen für das *RtOS Virtual Network* vorzunehmen, doppelklicken Sie in den Netzwerkverbindungen auf den gleichnamigen Eintrag (2) (Bild 14) und öffnen das zugehörige Eigenschaftsfenster. Dort wählen Sie *Internet Protokoll (TCP/IP)* und klicken auf den Button *Eigenschaften* (Bild 17).

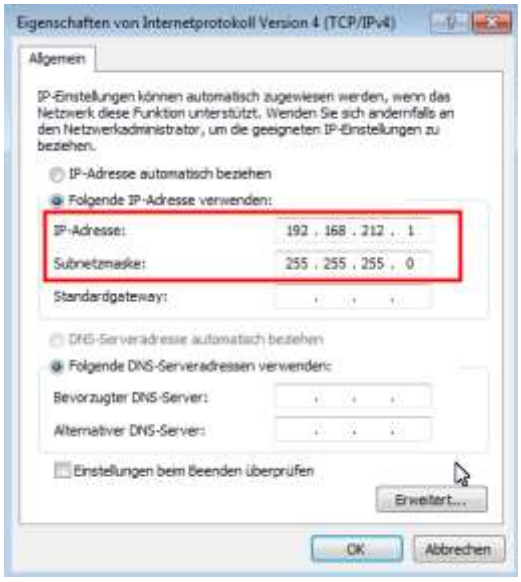
Bild 17:
Eigenschaften von RtOS Virtual Network



Im nächsten Fenster *Eigenschaften von Internet Protocol (TCP/IP)* übernehmen Sie die vorgegebenen Werte (Bild 18):

- IP-Adresse: 192.168.212.1
- Subnet-Mask: 255.255.255.0
- Standardgateway: leer
- DNS-Serveradressen: leer

Bild 18:
Eintragen von IP-Adresse und Subnetzmaske für RtOS Virtual Network


Eigenschaften von Internetprotokoll Version 4 (TCP/IPv4)
Allgemein
IP-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IP-Einstellungen zu beziehen.
IP-Adresse automatisch beziehen
Folgende IP-Adresse verwenden:
IP-Adresse: 192 . 168 . 212 . 1
Subnetzmaske: 255 . 255 . 255 . 0
Standardgateway:
DNS-Serveradresse automatisch beziehen
Folgende DNS-Serveradressen verwenden:
Bevorzugter DNS-Server:
Alternativer DNS-Server:
Einstellungen beim Beenden überprüfen
Erweitert...
OK
Abbrechen

IP-Adressen ändern mit X-Manager

Siehe Kapitel Schleicher X-Manager Seite 123.

6.2.5 Kommunikation zur Programmierung mit MULTIPROG (Schritt 4)

Für die Kommunikation mit einem externen PC zur Programmierung mit MULTIPROG über das Hausnetz muss zuerst eine passende Netzwerkroute eingerichtet werden.

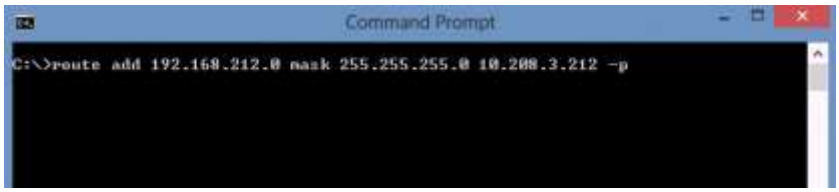
Die Eingabe der Parameter erfolgt unter *Start / Programme / Zubehör / Eingabeaufforderung* (Bild 19).

Beispiel:

```
route add 192.168.212.0 mask 255.255.255.0
10.208.3.212 –p
```

(-p für permanent), mit <Enter> übernehmen.

Bild 19: Einrichten der Netzwerkroute zur Programmierung mit MULTIPROG



Mit dem PING-Befehl können Sie anschließend einen Test ausführen, um zu sehen, ob die Kommunikationsverbindung besteht (die XCA Steuerung muss angeschlossen und gestartet sein):

`ping 192.168.212.2` *<Enter>*

Anzeige bei korrekter Verbindung:

```
Antwort von 192.168.212.2: Bytes=32 Time<1ms TTL=63
Antwort von 192.168.212.2: Bytes=32 Time<1ms TTL=63
Antwort von 192.168.212.2: Bytes=32 Time<1ms TTL=63
Antwort von 192.168.212.2: Bytes=32 Time=1ms TTL=63

Ping-Statistik für 192.168.99.2:
   Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
   (0% Verlust),

Ca. Zeitangaben in Millisek.:
   Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms
```

Auf dem externen PC kann jetzt MULTIPROG gestartet werden. Wählen Sie *Neues Projekt / XCA11xx* (oder öffnen Sie ein bereits bestehendes Projekt mit der vorhandenen Steuerung). Eine ausführliche Beschreibung der Vorgehensweise finden Sie im Kapitel " Erste Schritte mit MULTIPROG" auf Seite 42.

Im SPS-Projekt klicken Sie mit der rechten Maustaste auf den Eintrag *Resource : XCx11* und wählen im Kontextmenü *Einstellungen* (Bild 20).

Bild 20:
Aufruf der Ressource-Einstellungen in MULTIPROG

Im Dialogfenster *Ressource-Einstellungen für XCx11* wird die IP-Adresse angegeben (Bild 21).

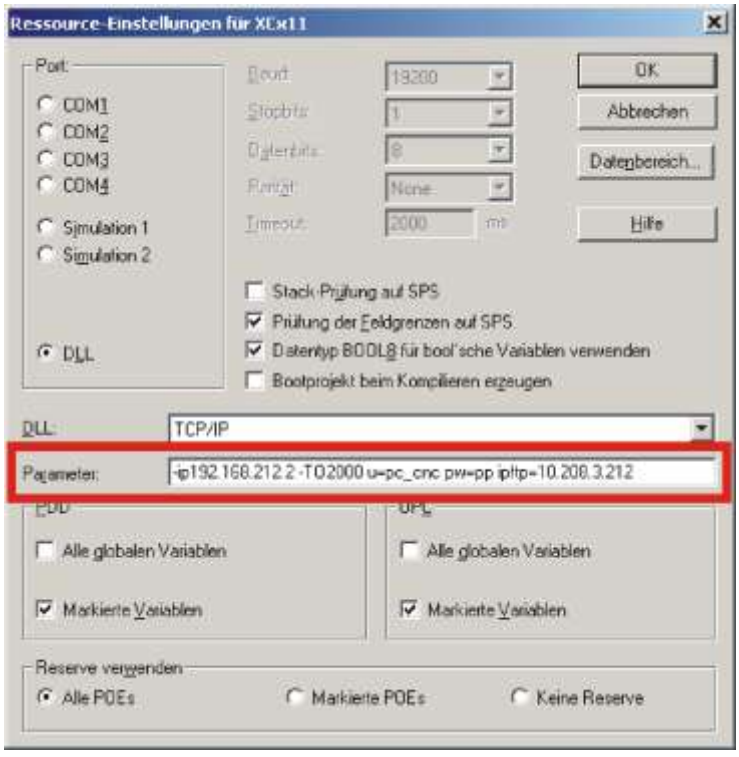
Beim Anlegen eines neue Projekts sehen Sie als Template folgende Zeile: vxTarget -TO2000 u=pc_cnc pw=pp ipftp=vxHost.

Diese Alias-Adressen können auch beibehalten werden, dann müssen Sie aber in der Host-Datein unter Windows die entsprechenden IP-Adressen zuweisen.

Empfohlen wird aber ein Eintrag wie im folgenden Beispiel:

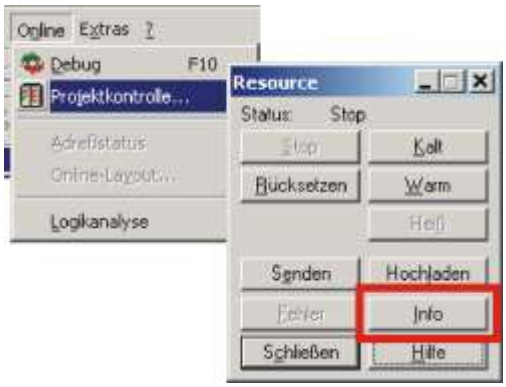
```
-ip192.168.212.2 -TO2000 u=pc_cnc pw=pp
ipftp=10.208.3.212
```

Bild 21:
Eingeben der IP-Adresse in MULTIPROG



Nach Bestätigen mit *OK* wird die Verbindung zur XCA Steuerung mit *Online / Projektkontrolle* und anschließend im Fenster *Resource* mit dem Button *Info* aufgerufen (Bild 22).

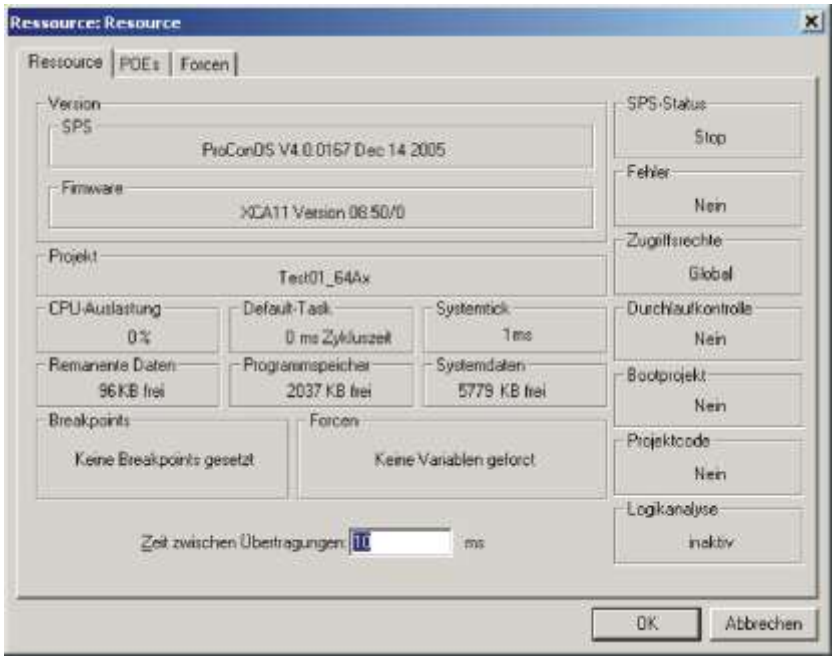
Bild 22:
Aufruf der Verbindung zur XCA Steuerung in MULTIPROG



Mit erfolgreicher Verbindung wird das Fenster ´Ressource:Resource´ angezeigt (Bild 23). Bei fehlerhafter Verbindung wird *Timeout* gemeldet. In diesem Falle sind alle Einstellungen nochmals zu überprüfen oder der Netzwerkadministrator zu Rate zu ziehen.

Eventuell sind weitere Einstellungen bei Gateways zur Subnetzvermittlung etc. zu beachten.

Bild 23:
Ressource-Fenster bei Online-Verbindung zur XCA Steuerung

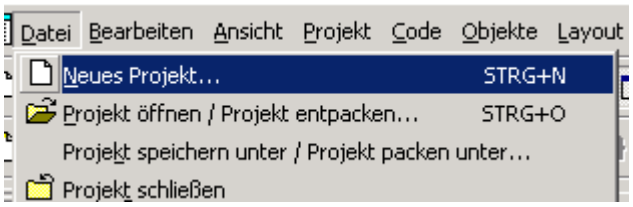


6.3 Erste Schritte mit MULTIPROG

6.3.1 MULTIPROG starten, neues Projekt öffnen und speichern

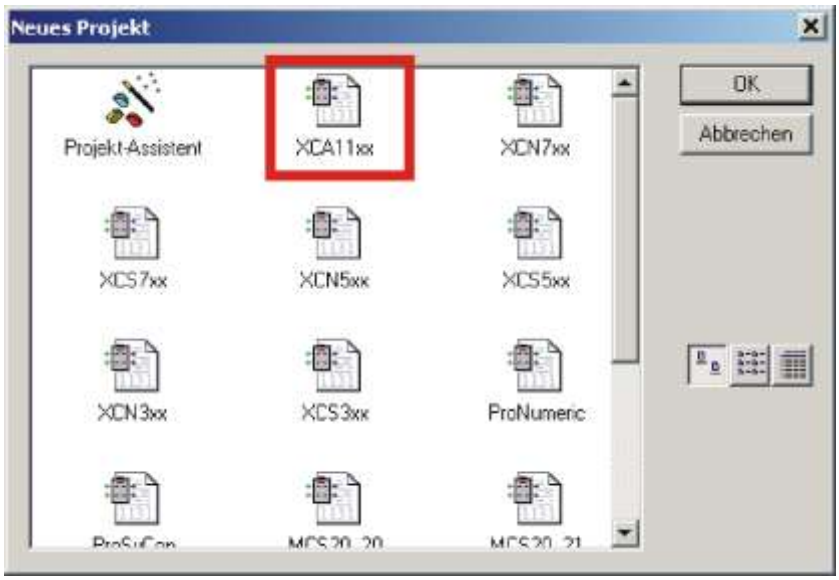
MULTIPROG starten, *Datei / Neues Projekt* wählen (Bild 24).

Bild 24:
Neues Projekt in MULTIPROG öffnen



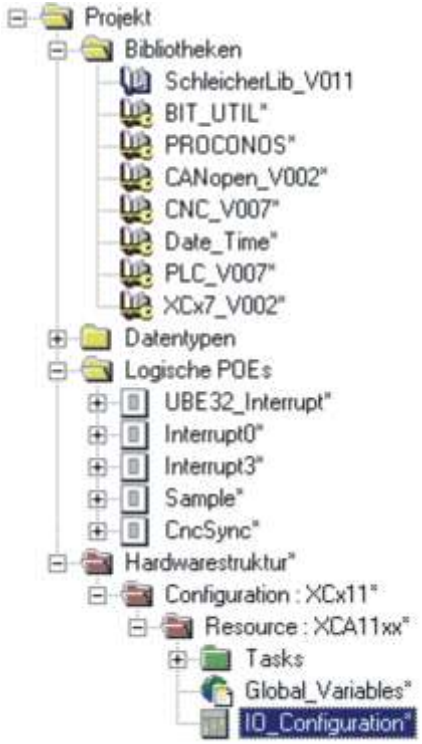
Ein Projekt für den vorhandenen Steuerungstyp (hier XCA 11xx) auswählen, mit *OK* ausführen (Bild 25).

Bild 25:
Steuerungstyp für neues Projekt auswählen



Ist das Projekt erfolgreich geöffnet, wird im Projektfenster der Projektbaum dargestellt (Bild 26). Es sind bereits Logische POEs enthalten, die voll funktionsfähig sind und für eine einfache Schnellinbetriebnahme ausreichen.

Bild 26:
Beispiel für Projektbaum in MULTIPROG



Unter *Datei / Projekt speichern unter* das Projekt mit einem neuen Namen (hier START) speichern (Bild 27 und Bild 28).

Bild 27:
Speicherdialog aufrufen

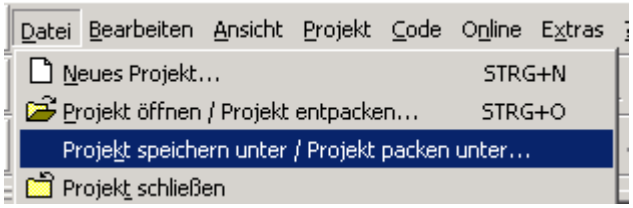
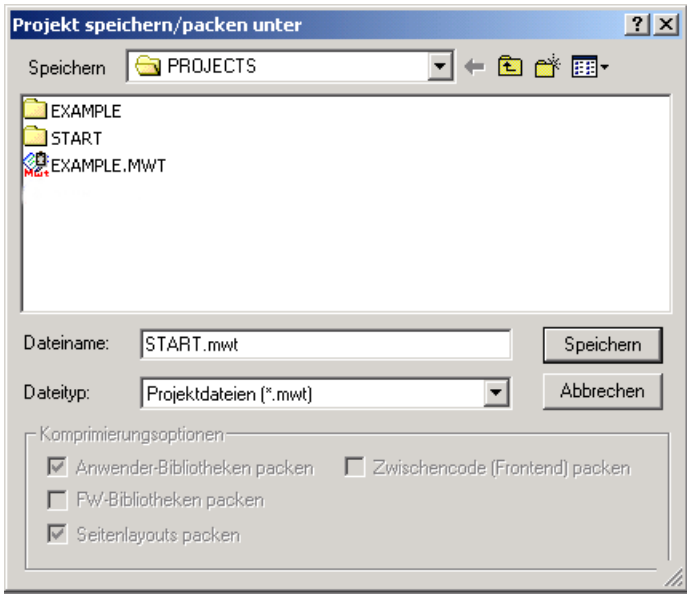

Datei Bearbeiten Ansicht Projekt Code Online Extras
Neues Projekt... STRG+N
Projekt öffnen / Projekt entpacken... STRG+O
Projekt speichern unter / Projekt packen unter...
Projekt schließen

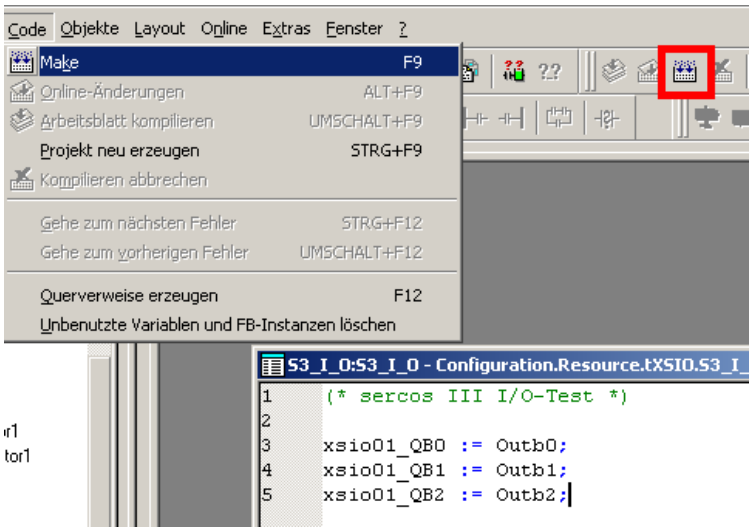
Bild 28:
Dialogfenster "Projekt speichern"



6.3.2 Ein Projekt kompilieren und zur XCx senden

Um das Projekt zu kompilieren, *Code / Make* wählen (Bild 29) (alternativ <F9> oder Button *Make*).

Bild 29:
Projekt kompilieren



Im Meldungsfenster unten wird der Kompiler-Lauf protokolliert. Hier werden auch Fehlermeldungen, Warnungen und weitere Infos angezeigt. Ein Anwählen der Auswahlbox "Fehler", "Warnungen" usw. zeigt die Meldung detaillierter an.

Werden Fehler angezeigt, kann mit Doppelklick auf der Fehlerzeile direkt in die Zeile des SPS-Programms, die den Fehler verursachte, verzweigt werden.

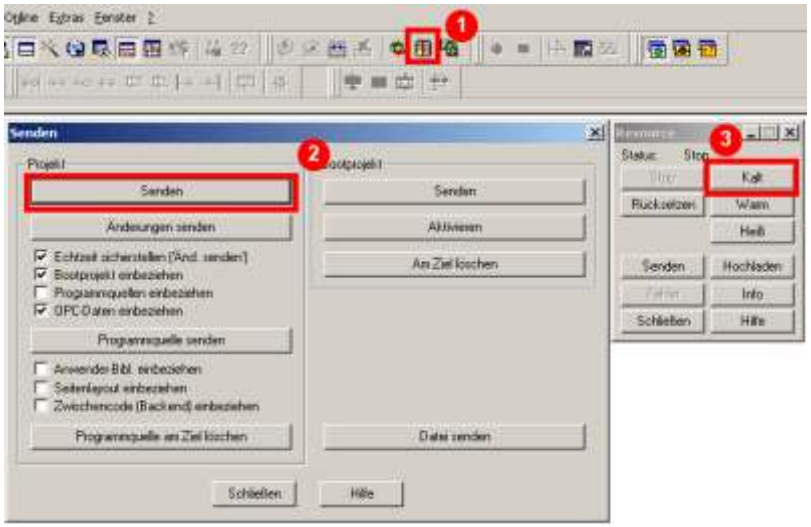
Anschließend übertragen Sie das Projekt über die Ethernet-Verbindung auf die Steuerung (Bild 30):

(1) Auf den Button *Projekt-Kontrolldialog* . Im sich daraufhin öffnenden Kontrolldialog *XCx* auf den Button *Senden* klicken.

(2) In der Rubrik *Projekt* wieder *Senden* wählen, damit wird das auf der XCx vorhandene Projekt überschrieben.

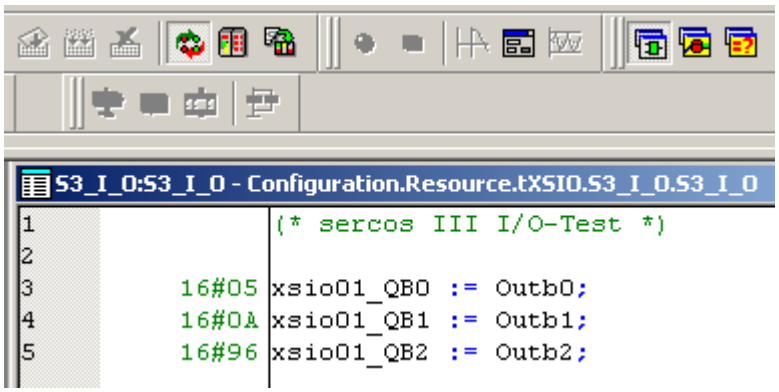
(3) Mit dem Button *Kalt* (Kaltstart) im Kontrolldialog *Resource* wird das Programm auf der XCx gestartet. Die LED *PLC RUN* auf der Steuerung leuchtet dauernd grün.

Bild 30:
Projekt auf die Steuerung übertragen



Mit dem Button *Debug ein/aus* kann auf dem Arbeitsblatt der Inhalt der Variablen online angezeigt werden (Bild 31).

Bild 31:
Online-Anzeige der Variablen


S3_I_O:S3_I_O - Configuration.Resource.tXSIO.S3_I_O.S3_I_O
1 (* sercos III I/O-Test *)
2
3 16#05 xsio01_QB0 := Outb0;
4 16#0A xsio01_QB1 := Outb1;
5 16#96 xsio01_QB2 := Outb2;

6.4 Einfügen des Koppelspeichers

Um den vollen Zugriff auf alle vordefinierten Variablen zu erlangen, ist es an dieser Stelle der Inbetriebnahme angebracht, die Koppelspeicherstruktur in das Projekt einzufügen.

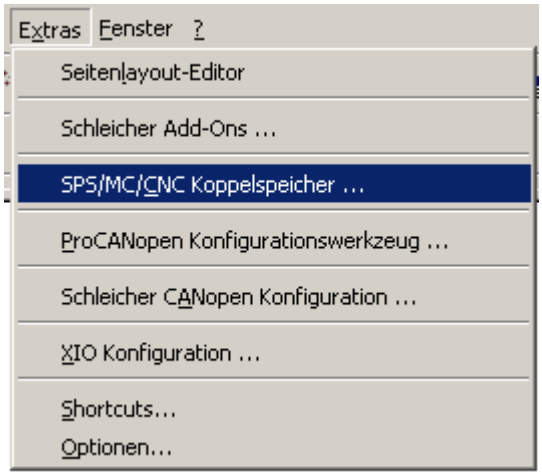
Der Koppelspeicher besitzt eine umfangreiche Datenstruktur, die zur Kommunikation zwischen SPS, CNC und Visualisierungssystemen verwendet wird. Außerdem kann über den Koppelspeicher auf Datenbereiche wie z.B. Version Number, Error Page und Log Book zugegriffen werden.

Eine Einführung zum Thema Koppelspeicher finden Sie auf Seite 117. Eine genaue Beschreibung sämtlicher Variablen des Koppelspeichers ist als Online-Hilfe der Software "Schleicher-Dialog" verfügbar. Diese kann auch über das Hilfe-Menü von MULTIPROG aufgerufen werden.

Um auf Variablen des Koppelspeichers zugreifen zu können, müssen die entsprechenden Datentypen und Variablen in das SPS-Projekt eingefügt werden. Diese sind in den Projekt-Templates von MULTIPROG noch nicht enthalten. Sie müssen vom Anwender selbst mit Hilfe des Koppelspeicher-AddOns für MULTIPROG in das SPS-Projekt eingefügt werden. Damit soll sichergestellt werden, dass der Anwender mit der für die Steuerungssoftware passenden Koppelspeicher-Variante arbeitet.

Das Einfügen der Koppelspeicher-Datentypen und -Variablen in ein SPS-Projekt erfolgt über das Menü *Extras / SPS/MC/CNC Koppelspeicher* (Bild 32).

Bild 32:
Aufruf des Dialogfensters "SPS/MC/CNC Koppelspeicher"

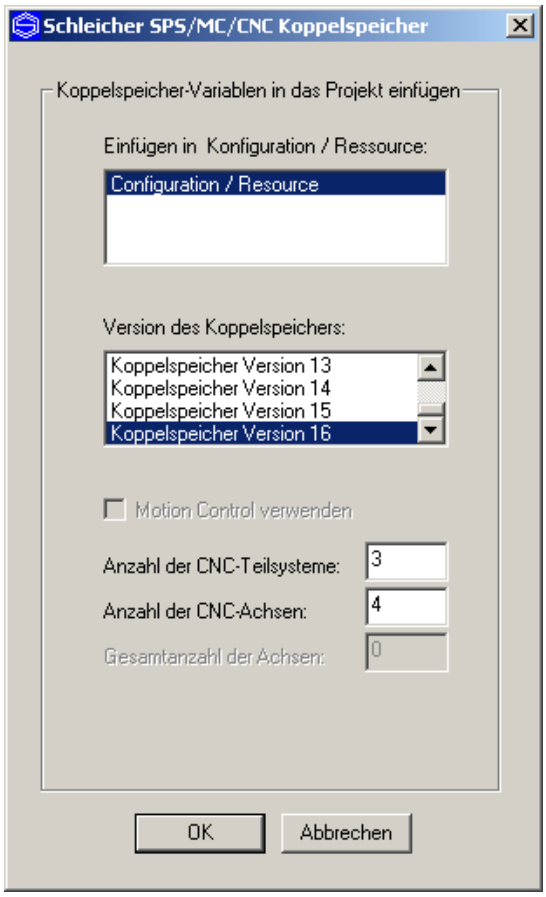


Es öffnet sich ein Dialogfenster mit folgenden Auswahl- bzw. Eingabemöglichkeiten (Bild 33).

- *Einfügen in Konfiguration / Ressource*:
  Auswahl der Ressource des SPS-Projektes, in die die Koppelspeicher-Variablen eingefügt werden sollen.
- *Version des Koppelspeichers*:
  Auswahl der Koppelspeicher-Version.
- *Anzahl der CNC-Teilsysteme / CNC-Achsen*:
  Eingabe der Anzahl der Teilsysteme und Achsen für CNC-Steuerungen (für reine SPS-Steuerungen sind diese beiden Eingabefelder deaktiviert).
- *OK-Button*:
  Beim Verlassen des Dialogfensters mit OK werden die Koppelspeicher-Datentypen und -Variablen in das SPS-Projekt eingefügt.

- *Abbrechen-Button*:
  Beim Verlassen des Dialogfensters mit Abbrechen wird das SPS-Projekt nicht verändert.

Bild 33: Dialogfenster "SPS/CNC/MC Koppelspeicher"



Wichtig!

Die gewählte Koppelspeicher-Version muss zur Version des Steuerungsbetriebssystems passen (siehe "Hinweise zur Auswahl der Koppelspeicher-Version", Seite 48.)

Nach dem Einfügen der Koppelspeicher-Datentypen und -Variablen muss das SPS-Projekt neu übersetzt und zur Steuerung übertragen werden.

6.4.1 Zugriff auf den Koppelspeicher

Das SPS-Programm hat Zugriff auf den gesamten Koppelspeicher über die globale Variable `plcMem` (bei SPS-Steuerungen) bzw. `cncMem` (bei CNC-Steuerungen). Mit der Punktschreibweise kann auf die einzelnen Komponenten des Koppelspeichers zugegriffen werden. Zum Beispiel kann das SPS-Programm die Versionsnummer der Betriebssoftware der Steuerung folgendermaßen auslesen: `cncMem.plcSect.lOSVersion`.

Visualisierungssysteme haben Zugriff auf den Koppelspeicher über die OPC-Schnittstelle. Die Versionsnummer der Betriebssoftware kann beispielsweise aus der OPC-Variablen `cmpS_lOSVersion` ausgelesen werden.

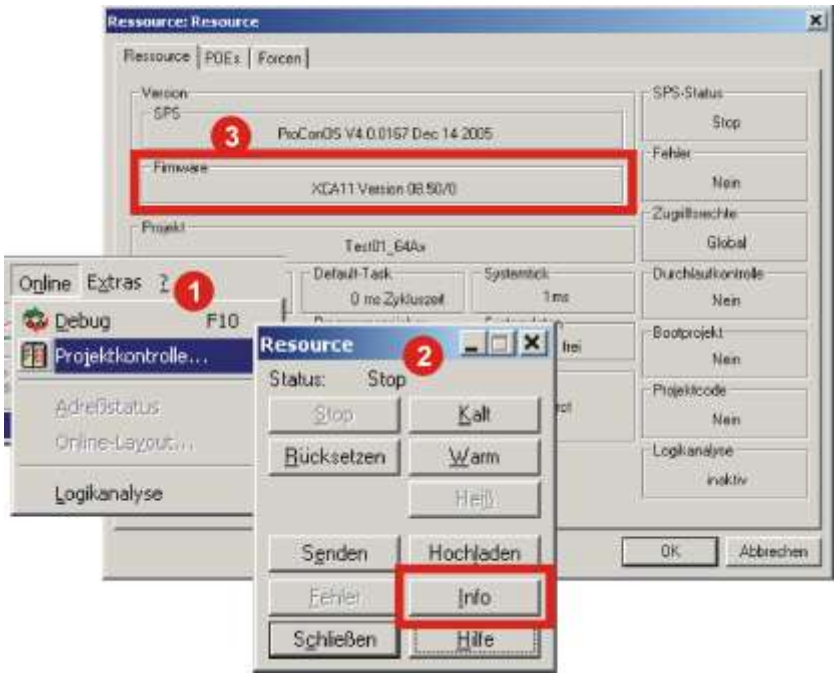
6.4.2 Hinweise zur Auswahl der Koppelspeicher-Version

Die Koppelspeicher-Struktur wird von Schleicher von Zeit zu Zeit aktualisiert bzw. erweitert. Zur Unterscheidung der einzelnen Varianten dient eine Versionsnummer. Die Versionsnummer wird bei größeren Änderungen an der Koppelspeicher-Struktur erhöht, bei der eine Änderung von Adressen von Variablen erforderlich ist. Koppelspeicher-Varianten mit unterschiedlicher Versionsnummer sind daher inkompatibel. Nur bei übereinstimmender Versionsnummer ist eine Kompatibilität gewährleistet.

Grundsätzlich sollte immer die aktuellste Koppelspeicher-Version verwendet werden. Wenn es sich jedoch um eine Steuerung mit einer älteren Betriebssoftware handelt, muss eine hierzu passende ältere Koppelspeicher-Version verwendet werden. Das Dokument "XCx11xx Revision History" gibt Auskunft darüber, welche Koppelspeicher-Version mit welcher Betriebssoftware-Version verwendet werden kann. Das Dokument ist auf der Schleicher-Homepage http://www.schleicher-electronic.com unter "Betriebsanleitungen XCx" verfügbar oder wird auf Anfrage bereitgestellt.

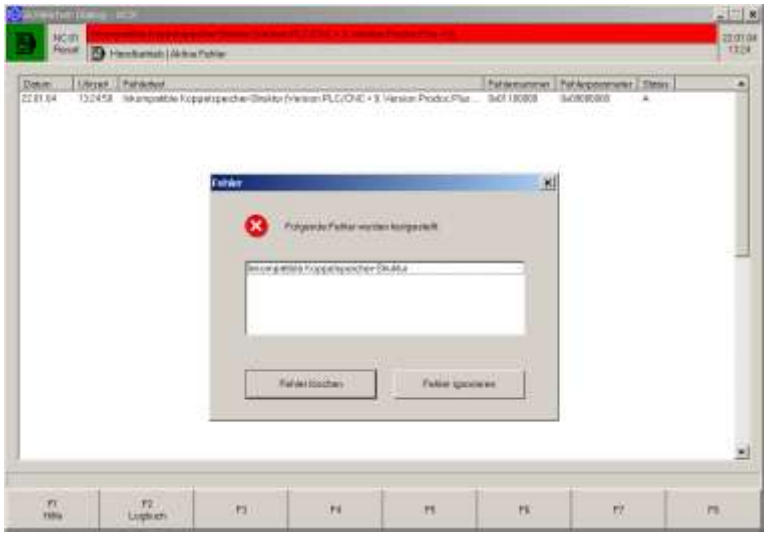
Die Version der Betriebssoftware der Steuerung kann im Info-Dialogfenster (3) zu der entsprechenden Ressource im SPS-Projekt angezeigt werden (über (1) *Online / Projektkontrolle*, dann (2) *Resource / Info*) (Bild 34).

Bild 34:
Auslesen der Betriebsoftware-Version



Wenn die Betriebssoftware der Steuerung und die verwendete Koppelspeicher-Version nicht übereinstimmen, wird beim Start der SPS eine Fehlermeldung in den Fehlerspeicher eingetragen. Diese Fehlermeldung wird im Schleicher-Dialog folgendermaßen angezeigt (Bild 35).

Bild 35: Darstellung der Koppelspeicher-Fehlermeldung im Schleicher-Dialog

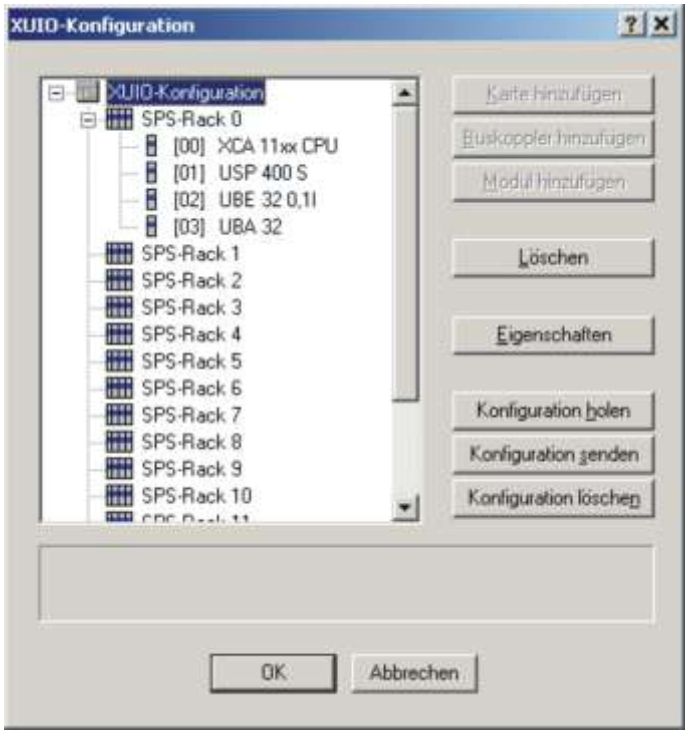


Falls eine solche Fehlermeldung auftritt, muss das SPS-Projekt korrigiert werden, indem die Datentypen und Variablen für die passende Koppelspeicher-Version eingefügt werden. Anschließend muss das SPS-Projekt neu übersetzt und zur Steuerung übertragen werden.

6.5 Zugriff auf die I/O-Ebene

6.5.1 Die Buttons im Dialogfenster XUIO-Konfiguration

Bild 36:
Dialogfenster "XUIO-Konfiguration"



Karte hinzufügen; Buskoppler hinzufügen

nicht relevant für XCA Steuerung.

Modul hinzufügen

Hinzufügen eines neuen Moduls. Es erscheint eine Auswahlliste mit allen Modulen für die XCA Steuerung. Anschließend können in einem weiteren Dialogfenster die Eigenschaften des Moduls (Steckplatznummer, Option, Moduladressen) festgelegt werden. Nach Bestätigung mit *OK* wird das Modul hinzugefügt. Es wird entsprechend der gewählten Steckplatznummer in die Hardware-Konfiguration einsortiert. Ein neues Modul kann nur dann hinzugefügt werden, wenn in der Baumansicht ein SPS-Rack oder ein anderes Modul ausgewählt wurde.

Löschen

Löschen eines oder mehrerer Module. Wenn in der Baumansicht ein Modul ausgewählt wurde, wird dieses Modul aus der Hardware-Konfiguration entfernt. Wenn in der Baumansicht ein SPS-Rack ausgewählt wurde, werden alle zu diesem Rack gehörigen Module entfernt. Wenn in der Baumansicht der Knoten *XUIO-Konfiguration* ausgewählt wurde, werden sämtliche Module aus der Hardware-Konfiguration entfernt.

Eigenschaften

Anzeigen der Eigenschaften der Gesamtkonfiguration bzw. eines Moduls.

Wenn in der Baumansicht der Knoten *XUIO-Konfiguration* ausgewählt wurde, werden die Eigenschaften der Gesamtkonfiguration angezeigt. Hier kann festgelegt werden, welche Variablen für die digitalen Ein- und Ausgänge von Modulen generiert werden sollen und welche Einträge in die I/O-Konfiguration des SPS-Projektes eingefügt werden sollen.

Für SPS-Racks sind keine Eigenschaften verfügbar.

Wenn in der Baumansicht ein Modul ausgewählt wurde, werden die Eigenschaften dieses Moduls angezeigt. Hier können Steckplatz, Option und Moduladressen geändert werden.

Konfiguration holen

Aktuelle Hardware-Konfiguration von der Steuerung holen und in der Baumansicht anzeigen. Diese Funktion kann nur ausgeführt werden, wenn eine Verbindung zur Steuerung hergestellt werden kann.

Konfiguration senden

Die in der Baumansicht angezeigte Hardware-Konfiguration zur Steuerung übertragen. Diese Funktion kann nur ausgeführt werden, wenn eine Verbindung zur Steuerung hergestellt werden kann. Hiermit wird die erstellte Hardware-Konfiguration auf dem CompactFlash-Speicher der Steuerung gespeichert. Die vorherige Hardware-Konfiguration wird dadurch überschrieben. Damit die übertragene Hardware-Konfiguration wirksam wird, muss die Steuerung neu gestartet werden.

Konfiguration löschen

Hardware-Konfiguration auf der Steuerung löschen. Diese Funktion kann nur ausgeführt werden, wenn eine Verbindung zur Steuerung hergestellt werden kann. Hiermit wird die auf dem CompactFlash-Speicher der Steuerung gespeicherte Hardware-Konfiguration gelöscht.

OK

Abspeichern der erstellten Hardware-Konfiguration, Einfügen der Variablen und I/O-Konfigurations-Einträge in das SPS-Projekt entsprechend der erstellten Hardware-Konfiguration und Schließen des Fensters.

Abbrechen

Schließen des Fensters ohne Veränderungen am SPS-Projekt und ohne Speichern der erstellten Hardware-Konfiguration.

6.5.2 Einlesen der Hardware-Konfigurationen

Mit der Installation der AddOns für MULTIPROG wird ein Konfigurator-Werkzeug zur Verfügung gestellt, das folgende Aufgaben übernimmt:

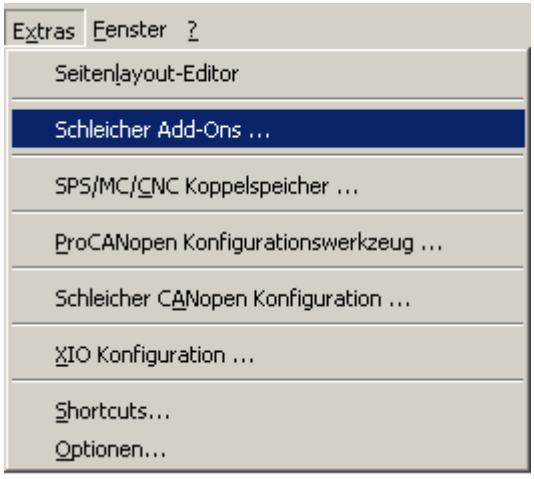
- Anzeigen und Editieren der Hardware-Konfiguration (Steckplatzliste der Eingangs- und Ausgangsmodule)
- Einfügen der erforderlichen Variablen in das SPS-Projekt, mit deren Hilfe der SPS-Programmierer auf die Eingangs- und Ausgangs-module zugreifen kann
- Einfügen der erforderlichen Einträge in die I/O-Konfiguration des SPS-Projektes

Hierfür stehen folgende I/O-Konfiguratoren zur Verfügung:

- Unter Schleicher Add-Ons den Konfigurator für sercos III I/O
- Unter XIO Konfiguration den Konfigurator für XUIO Module

Der Aufruf vom sercos III I/O Konfigurator erfolgt über den Menüeintrag *Extras /Schleicher Add-Ons*.

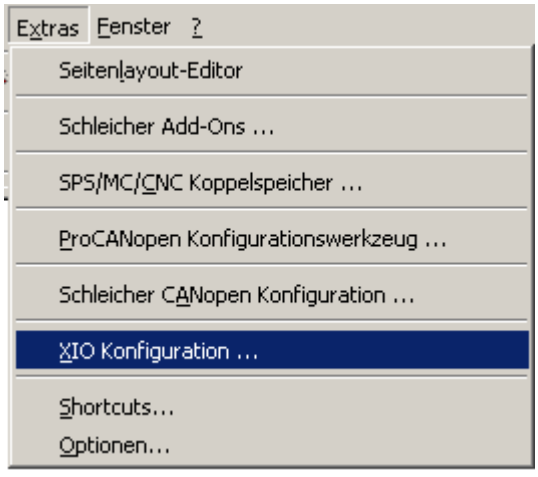
Bild 37: Menüeintrag "Extras / Schleicher Add-Ons"



Zur Inbetriebnahme der sercos III I/O siehe die Betriebsanleitung für sercos III I/O (siehe Zusätzliche Betriebsanleitungen Seite 10).

Der Aufruf vom XIO Konfigurator erfolgt über den Menüeintrag *Extras / XIO Konfiguration* gestartet werden (Bild 38).

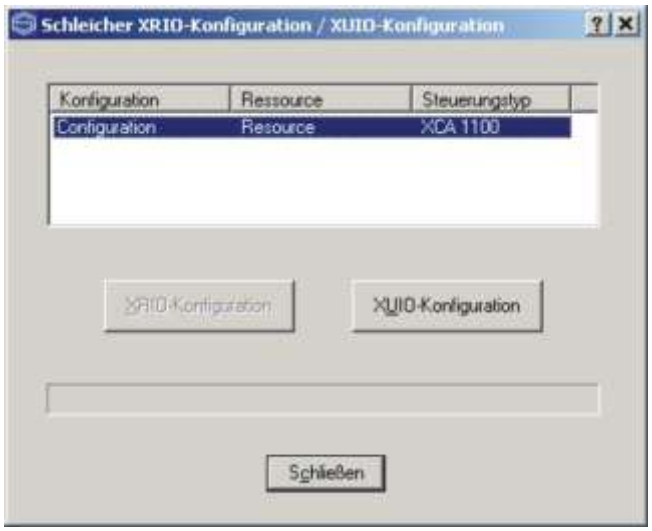
Bild 38: Menüeintrag "Extras / XIO Konfiguration"



Zunächst erscheint das Dialogfenster *XRIO-Konfiguration / XUIO-Konfiguration*, in dem die im SPS-Projekt vorhandenen Ressourcen

und die zugehörigen Steuerungstypen zur Auswahl angezeigt werden. Mit dem Button *XUIO-Konfiguration* wird die Konfigurierung für die ausgewählte Ressource fortgesetzt (Bild 39).

Bild 39:
Dialogfenster "XRIO-Konfiguration / XUIO-Konfiguration"

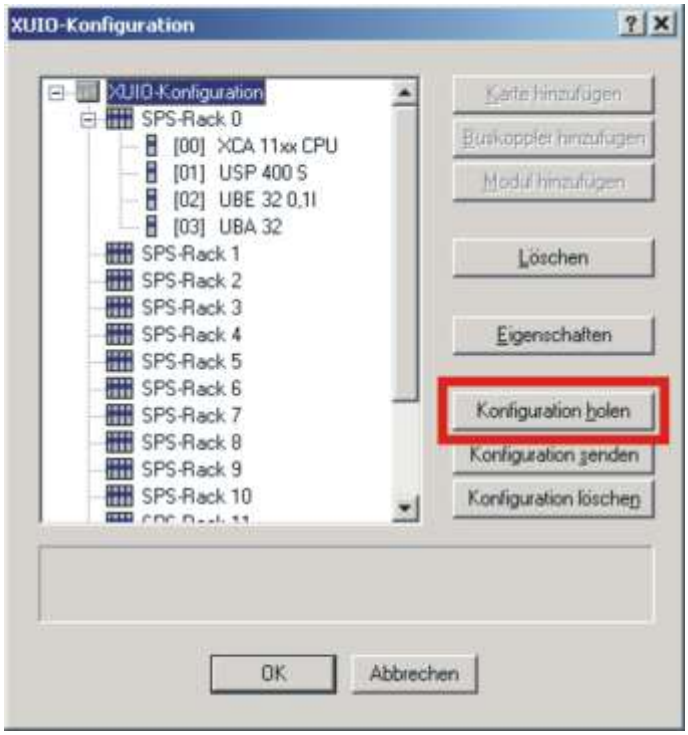


Wichtig!

Treten Fehler bei der Kommunikation über die Ethernet-Verbindung zur XCx auf, muss die korrekte Parametrierung der Schnittstelle in MULTIPROG überprüft werden.

Im Dialogfenster wird die im SPS-Projekt gespeicherte Hardware-Konfiguration angezeigt. Bei einem neuen SPS-Projekt ist die Hardware-Konfiguration beim erstmaligen Öffnen dieses Fensters zunächst leer. In diesem Fall muss die Hardware-Konfiguration nur einmalig aus der Steuerung ausgelesen (Button *Konfiguration holen*) und mit *OK* in das SPS-Projekt eingefügt werden (Bild 40).

Bild 40:
Dialogfenster
"XUIO-Konfiguration"



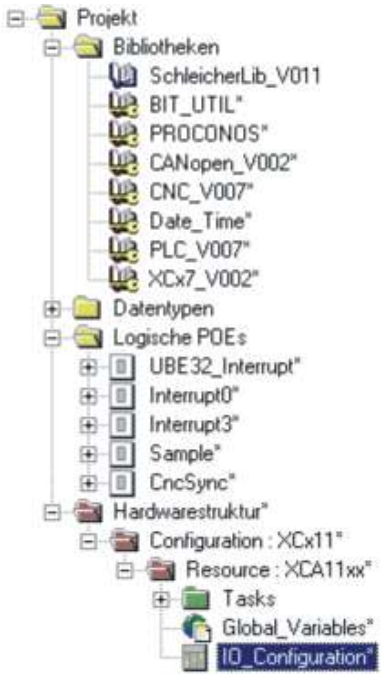
Die gespeicherte Hardware-Konfiguration wird bei jedem Steuerungsanlauf mit der tatsächlich vorhandenen Hardware-Konfiguration verglichen. Treten dabei Differenzen auf, wird der Steuerungsanlauf mit einer Fehlermeldung unterbrochen. Die LED *BUS* auf der Steuereinheit blinkt rot, im Active-Error-Buffer wird eine Fehlermeldung gespeichert:

0x01100001	Inkompatible Hardware-Konfiguration

Active-Error-Buffer und Error-Log-Book sind im Schleicher-Dialog auf jeder Bedienebene über die Tastenkombination <Ctrl+?> einsehbar.

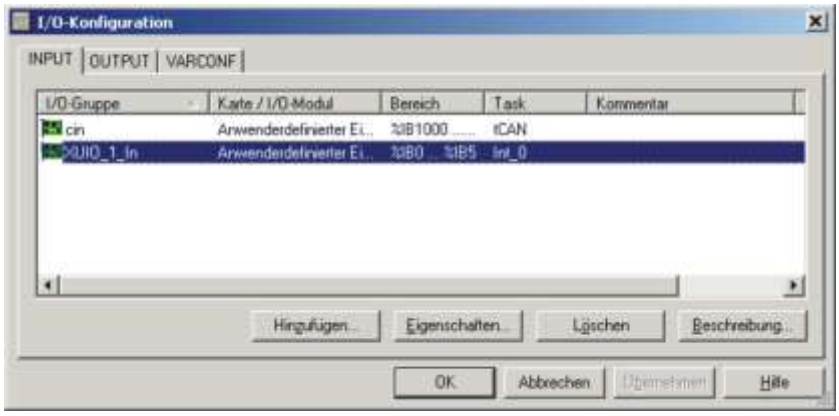
Die für die erstellte Hardware-Konfiguration erforderlichen Einträge in die I/O-Konfiguration werden in das SPS-Projekt eingefügt (die I/O-Konfiguration kann durch Doppelklick auf den Knoten *IO_Configuration* im Projektbaum geöffnet werden (Bild 41).

Bild 41: Knoten "IO_Configuration"



Für das oben gezeigte Beispiel werden die Einträge *XUIO_1_In* (unter INPUT) und *XUIO_1_Out* (unter OUTPUT) in die I/O-Konfiguration eingefügt (Bild 42).

Bild 42: Dialogfenster "I/O-Konfiguration"



Die für die erstellte I/O-Konfiguration erforderlichen Einträge in die Variablen-Tabelle werden in das SPS-Projekt eingefügt (Bild 44). Die Variablen-Tabelle kann durch Doppelklick auf den Knoten *Global_Variables* im Projektbaum geöffnet werden (Bild 43).

Bild 43: Knoten "Global_Variables"

XUIO_Variables				
xuio02_IW0	WORD	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, bit inputs	%IW 0
xuio02_IW2	WORD	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, bit inputs	%IW 2
xuio02_IW4	WORD	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, bit inputs	%IW 4
xuio02_QW0	WORD	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, bit outputs	%QW 0
xuio02_MWQ0	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000000
xuio02_MWQ2	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000002
xuio02_MWQ4	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000004
xuio02_MWQ6	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000006
xuio02_MWQ8	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000008
xuio02_MWQ10	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000010
xuio02_MWQ12	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000012
xuio02_MWQ14	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000014
xuio02_MWQ16	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000016
xuio02_MWQ18	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000018
xuio02_MWQ20	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000020
xuio02_MWQ22	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000022
xuio02_MWQ24	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000024
xuio02_MWQ26	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000026
xuio02_MWQ28	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000028
xuio02_MWQ30	INT	VAR_GLOBAL	XUIO slot 02, UBE 32 0,1I, word output	%MW 3.4000030
xuio03_QW2	WORD	VAR_GLOBAL	XUIO slot 03, UBA 32, bit outputs	%QW 2
xuio03_QW4	WORD	VAR_GLOBAL	XUIO slot 03, UBA 32, bit outputs	%QW 4

Bild 44: Einfügen der globalen Variablen in das SPS-Projekt

Für das obige Beispiel wurden in diesem Fall für die digitalen Ein- und Ausgänge Variablen vom Typ WORD erstellt. Dabei entspricht jedes der 16 Bits einer solchen Variablen einem digitalen Eingang bzw. Ausgang. Es kann jedoch auf Wunsch für jeden digitalen Ein- bzw. Ausgang auch eine eigene Variable vom Typ BOOL generiert werden. Hierzu muss die Option *Generierung von BOOL-Variablen für digitale Ein-/Ausgänge* in den Eigenschaften der Gesamtkonfiguration aktiviert werden.

6.5.3 Einstellen von Hardware-Konfigurations-Optionen

Hinweis

Die folgenden Ausführungen sind nur für fortgeschrittene Benutzer wichtig.

Hardware-Konfigurations-Optionen (im folgenden Optionen genannt), erlauben eine effektive SPS-Programmerstellung. Ein SPS-Programm kann sich durch Abfrage der Optionen ohne Programmänderung an unterschiedliche Hardware-Konfigurationen anpassen.

Beispiel

Der I/O-Konfigurator wird erneut über den Menüeintrag *Extras / XIO Konfiguration* gestartet. Dann wird die oben eingelesene Hardware-Konfiguration über den Button *Modul hinzufügen* (1) mit zwei Zähler-Modulen UZB 2VR ergänzt (2) (Bild 45).

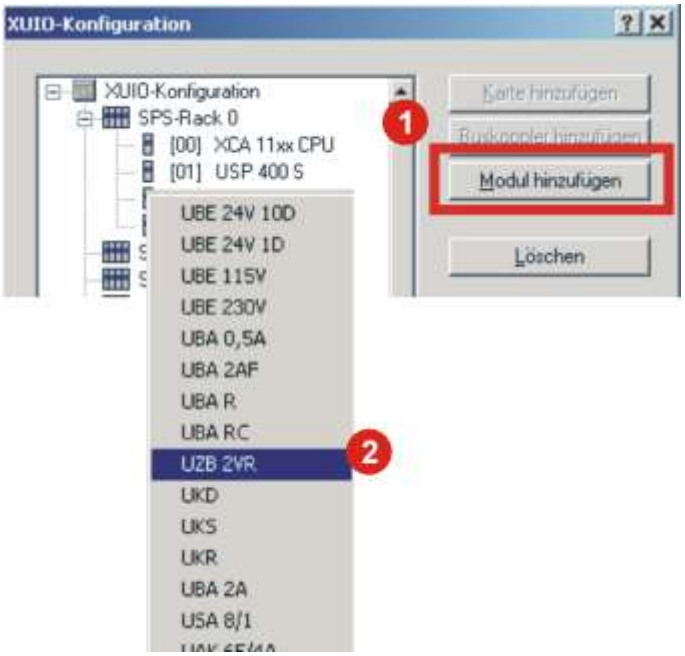
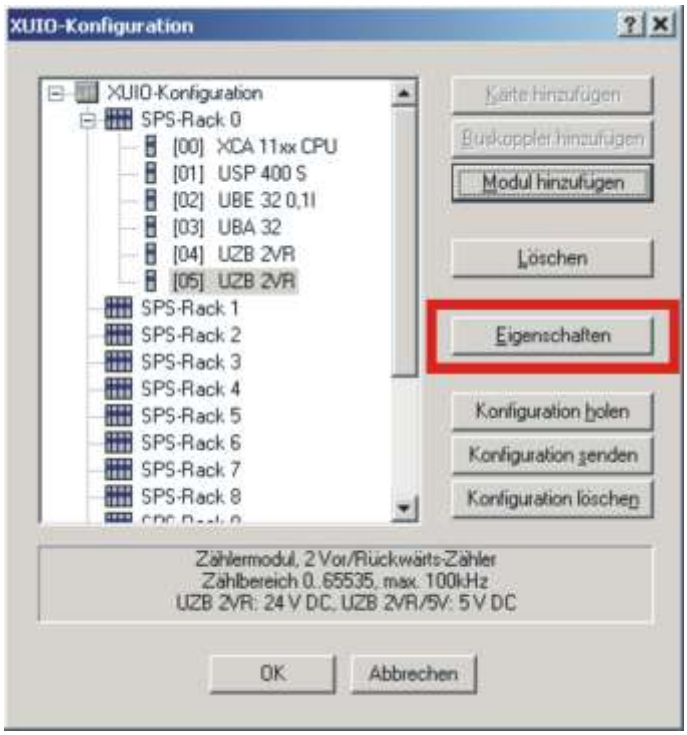


Bild 45: Dialogfenster "XUIO-Konfiguration", Modul hinzufügen

Anschließend wird für jedes der beiden neuen Module über den Button *Eigenschaften* das Dialogfenster *Modul-Eigenschaften* aufgerufen (Bild 46).

Bild 46: Dialogfenster "XUIO-Konfiguration", Eigenschaften



Für beide Module wird die Einstellung *Option* im Dialog *Modul-Eigenschaften* auf 1 eingestellt (Bild 47). Es sind insgesamt 32 Optionen 0...31 einstellbar. Alle anderen Module behalten ihre Einstellung, voreingestellt ist Option 0.

Bild 47: Dialogfenster "Modul-Eigenschaften", Einstellen der Optionen

Die beiden UZB 2VR gehören damit zur Option 1, alle anderen Module zur Option 0.

Bei Steuerungsanlauf wird die gespeicherte Hardware-Konfiguration mit der tatsächlich vorhandenen Hardware verglichen. Von einer Option müssen dabei entweder alle Module vorhanden sein (die Option ist dann aktiv), oder es darf kein Modul dieser Option vorhanden sein (die Option ist dann nicht aktiv). Die Adressbereiche nicht aktiver Optionen bleiben reserviert.

Sind die Module einer Option nur teilweise vorhanden, wird ein Konfigurationsfehler erkannt. (LED BUS auf der Steuereinheit blinkt rot, im Active-Error-Buffer wird eine Fehlermeldung gespeichert.)

6.6 Zugriff auf Interrupt-Eingänge

Die XCA Steuerung kann vier Interrupts verarbeiten, die durch digitale Eingänge ausgelöst werden. Dazu werden die ersten vier Eingänge eines Eingangs-Modules UBE 32 0,1I verwendet.

6.6.1 Variablen für die Interruptverarbeitung

Bei der Hardware-Konfiguration des Steuerungsaufbaues mit einem UBE 32 0,1I (siehe oben) werden neben den Variablen der physikalischen Eingänge auch die zur Interruptverarbeitung benötigten Variablen angelegt (Bild 48).

XUIO_Variable				
xuio03_IW0	WORD	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, bit inputs	%IW 0
xuio03_IW2	WORD	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, bit inputs	%IW 2
xuio03_IW4	WORD	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, bit inputs	%IW 4
xuio03_QW0	WORD	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, bit outputs	%QW 0
xuio03_MWQ0	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000000
xuio03_MWQ2	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000002
xuio03_MWQ4	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000004
xuio03_MWQ6	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000006
xuio03_MWQ8	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000008
xuio03_MWQ10	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000010
xuio03_MWQ12	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000012
xuio03_MWQ14	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000014
xuio03_MWQ16	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000016
xuio03_MWQ18	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000018
xuio03_MWQ20	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000020
xuio03_MWQ22	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000022
xuio03_MWQ24	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000024
xuio03_MWQ26	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000026
xuio03_MWQ28	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000028
xuio03_MWQ30	INT	VAR_GLOBAL	XUIO slot 03, UBE 32 0,1I, word output	%MW 3.4000030
xuio04_QW2	WORD	VAR_GLOBAL	XUIO slot 04, UBA 0,5A, bit outputs	%QW 2

Bild 48: Anlegen der Variablen für die Interruptverarbeitung

Wichtige Variablen des UBE 32 0,1I für die Interruptverarbeitung

Eingänge	
xuio.._IW0	Eingangs-Bits 0...15 (Eingangs-Bits 0...3 mit Interrupt-Funktionalität)
xuio.._IW2	Eingangs-Bits 16...31
xuio.._IW4	Abbild der Eingangs-Bits 0...3 bei freigeschalteter Interrupt-Funktionalität (Diese Bits müssen über xuio.._MWQ0-14 bei der Interrupt-Freigabe zurückgesetzt werden, siehe unten)
	Bit 0 wird gesetzt bei steigender Flanke am Eingang 0
	Bit 1 wird gesetzt bei steigender Flanke am Eingang 1
	Bit 2 wird gesetzt bei steigender Flanke am Eingang 2
	Bit 3 wird gesetzt bei steigender Flanke am Eingang 3
	Bit 4 wird gesetzt bei fallender Flanke am Eingang 0
	Bit 5 wird gesetzt bei fallender Flanke am Eingang 1
	Bit 6 wird gesetzt bei fallender Flanke am Eingang 2
	Bit 7 wird gesetzt bei fallender Flanke am Eingang 3
Ausgänge	
xuio.._QW0	Maske zur Freischaltung der Interrupts
	Bit 0 Eingangs-Bit 0 mit steigender Flanke
	Bit 1 Eingangs-Bit 1 mit steigender Flanke
	Bit 2 Eingangs-Bit 2 mit steigender Flanke
	Bit 3 Eingangs-Bit 3 mit steigender Flanke
	Bit 4 Eingangs-Bit 0 mit fallender Flanke
	Bit 5 Eingangs-Bit 1 mit fallender Flanke
	Bit 6 Eingangs-Bit 2 mit fallender Flanke
	Bit 7 Eingangs-Bit 3 mit fallender Flanke
xuio.._MWQ0 bis xuio.._MWQ14	Quittieren des Interrupts
	xuio.._MWQ0 Eingangs-Bit 0 mit steigender Flanke
	xuio.._MWQ2 Eingangs-Bit 1 mit steigender Flanke
	xuio.._MWQ4 Eingangs-Bit 2 mit steigender Flanke
	xuio.._MWQ6 Eingangs-Bit 3 mit steigender Flanke
	xuio.._MWQ8 Eingangs-Bit 0 mit fallender Flanke
	xuio.._MWQ10 Eingangs-Bit 1 mit fallender Flanke
	xuio.._MWQ12 Eingangs-Bit 2 mit fallender Flanke
	xuio.._MWQ14 Eingangs-Bit 3 mit fallender Flanke

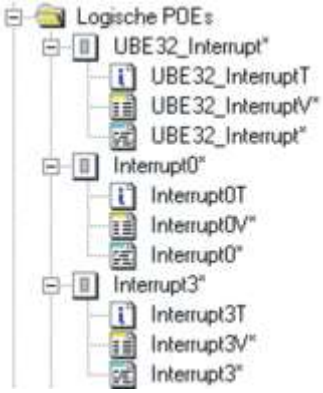
Tabelle 15: Wichtige Variablen des UBE 32 0,1I für die Interruptverarbeitung

Die Slot-Nummer des UBE 32 0,1I ist in der Tabelle mit zwei Punkten (xuio..) dargestellt.

6.6.2 Beispiel-POEs für die Interruptverarbeitung

Es werden drei POEs erstellt und in die Taskstruktur eingebunden. Die POEs dienen sowohl der Interrupt-Freigabe *UBE32_Interrupt* als auch der Interrupt-Verarbeitung eines Messtasters *Interrupt 0* und eines Zählimpulses *Interrupt3* (Bild 49).

Bild 49:
Knoten für Interrupt-freigabe und -verarbeitung



Es werden globale Variablen angelegt für die Zählung der aufgetretenen Interrupts und für die Kommunikation der POEs untereinander (Bild 50).

Bild 50:
Globale Variablen für Interruptzählung und Kommunikation

Interrupt0_Zaehler	INT	VAR_GLOBAL
Interrupt3_Zaehler	INT	VAR_GLOBAL
Interrupt0_RTrig_Freigabe	BOOL	VAR_GLOBAL

POE "UBE32_Interrupt"

```
(* Beispiel 1: POE Interrupt0/ Task I_E0 Ereignis 0
 Interrupt 0 (UBE32 0,1I Eingang 0)
 Auswertung der ersten steigender Flanke am Eingang (zB. ein Messtaster),
 Die Interrupt0_RTrig_Freigabe muss hier manuell durch Forcen gesetzt
 werden. Sie wird dann im Programm Interrupt0 zurückgesetzt
 ---------------------------------------------------------------- *)
IF    Interrupt0_RTrig_Freigabe
      THEN
            xuio03_QW0  := S_BIT_IN_WORD(TRUE,xuio03_QW0,SINT#0);
      ELSE
            xuio03_QW0  := R_BIT_IN_WORD(TRUE,xuio03_QW0,SINT#0);
            xuio03_MWQ0 := 0;
END_IF;

(* Beispiel 2: POE Interrupt3/ Task I_E3 Ereignis 3
 Interrupt 3 (UBE32 0,1I Eingang 3)
 Auswertung aller fallenden Flanken am Eingang (zB. Zehlen mit einer
 Lichtschranke). Die Interrupt3_RTrig_Freigabe muss manuell gesetzt und
zuruechgesetzt werden.
 ---------------------------------------------------------------- *)
IF    Interrupt3_FTrig_Freigabe
      THEN
            xuio03_QW0   := S_BIT_IN_WORD(TRUE,xuio03_QW0,SINT#7);
      ELSE
            xuio03_QW0   := R_BIT_IN_WORD(TRUE,xuio03_QW0,SINT#7);
            xuio03_MWQ14 := 0;
END_IF;
```

POE "Interrupt0" (Messtater)

```
(* Beispiel 1: POE Interrupt0/ Task I_E0 Ereignis 0
 Interrupt 0 (UBE32 0,1I Eingang 0)
 Auswertung der ersten steigender Flanke am Eingang als Messtaster
 ---------------------------------------------------------------- *)
Interrupt0_Zaehler        := Interrupt0_Zaehler + 1;

Interrupt0_RTrig_Freigabe := FALSE;

RETURN;
```

POE "Interrupt3" (Zähler)

```
(* Beispiel 2: POE Interrupt3/ Task I_E3 Ereignis 3
 Interrupt 3 (UBE32 0,1I Eingang 3)
 Auswertung aller fallenden Flanken am Eingang als Zaehler
 ---------------------------------------------------------------- *)
Interrupt3_Zaehler  := Interrupt3_Zaehler + 1;

xuio03_MWQ14   := 0;

RETURN;
```

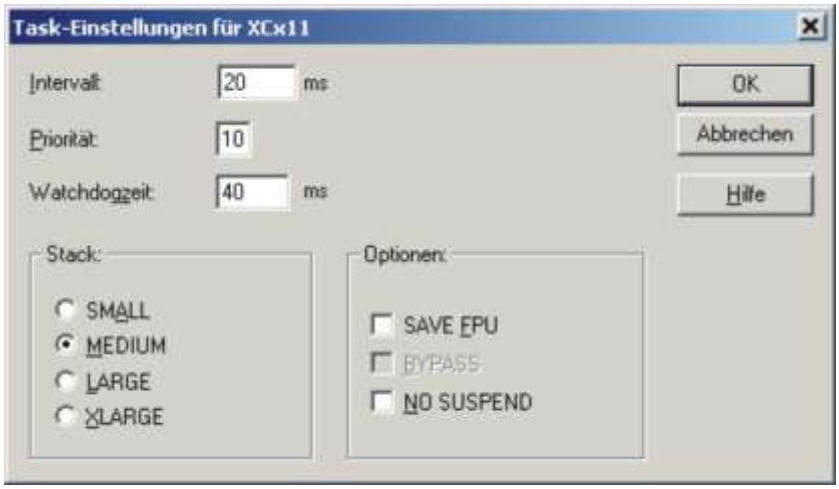
6.6.3 Taskstruktur für die Interruptverarbeitung

Für die Interruptfreigabe wird eine zyklische Task mit dem Namen *XUIOTsk* angelegt, der die POE *UBE32_Interrupt* zugewiesen wird (Bild 51).

Bild 51: Zyklische Task "XUIO" mit POE "UBE32_Interrupt"

Als Beispiel werden folgende Task-Einstellungen gewählt (Bild 52):

Bild 52: Einstellungen für zyklische Task



Zuletzt werden Event-Tasks angelegt, die interruptabhängig gestartet und denen die interrupverarbeitenden POEs zugewiesen werden (Bild 53).

Bild 53: Anlegen der Event-Tasks für die Interrupts

Hier wurden zwei Event-Tasks angelegt:

- *I_E0:EVENT* für Interrupt über Eingangs-Bit 0 mit POE *Interrupt0*
- *I_E3:EVENT* für Interrupt über Eingangs-Bit 3 mit POE *Interrupt3*

Die Zuweisung der Interrupts (Eingangs-Bits) zu den Event-Tasks muss in der Taskeinstellung mittels der Ereignisnummer angegeben werden (Bild 54). Dabei gilt die Festlegung:

Eingangs-Bit 0	Ereignis 0
Eingangs-Bit 1	Ereignis 1
Eingangs-Bit 2	Ereignis 2
Eingangs-Bit 3	Ereignis 3

Bild 54: Einstellungen für Event-Tasks

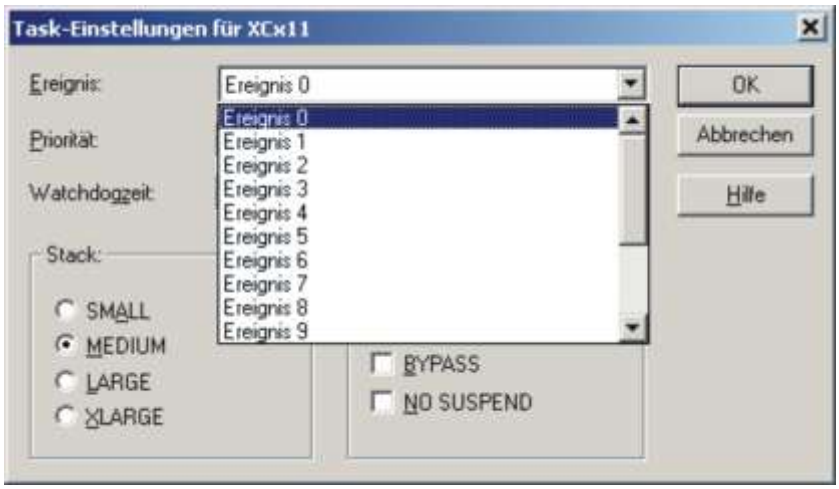
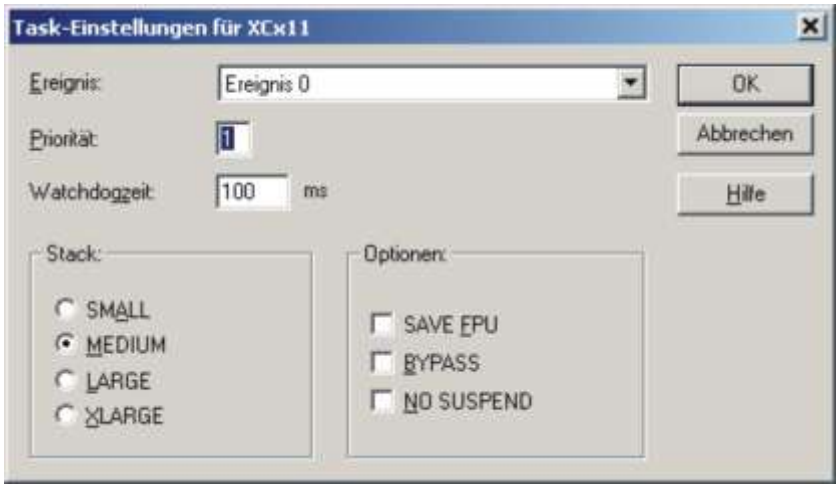


Bild 55: Einstellungen für Event-Tasks



6.7 CANopen für dezentrale I/O

Die XCA Steuerung besitzt eine integrierte CANopen-Schnittstelle (optional) für die Feldbusanbindung oder die Ansteuerung digitaler Antriebe. Dieses Kapitel beschreibt anhand einer Minimalkonfiguration die Inbetriebnahme und Konfiguration des CANopen-Netzwerkes.

6.7.1 Spezifikationen

- CANopen arbeitet mit zwei Typen von Telegrammen:
  - **SDO** (Service Data Objects) sind Telegramme, die vom Empfänger **bestätigt** werden müssen,
  - **PDO** (Process Data Objects) sind Telegramme, die vom Empfänger **nicht bestätigt** werden müssen.
- Während der Netzwerk-Konfiguration werden die PDOs für den Datenaustausch definiert und bekommen eine sogenannte COB ID. Der Empfänger einer Nachricht erkennt zu jeder Zeit, welches Telegramm für diesen Knoten bestimmt ist.
- Einige Komponenten unterstützen nur sogenanntes Default Mapping und arbeiten mit festen COB IDs, die in der CANopen-Definition festgeschrieben sind.
- Die Standardkommunikation für PDO ist **COS** (**C**hange **O**f **S**tate): Eine PDO wird nur gesendet, wenn sich die Information innerhalb der PDO ändert.

6.7.2 Anschlussprinzip und Verkabelung

Der beschriebene Aufbau ist eine Minimalkonfiguration, die für die weitere Inbetriebnahme als Beispiel dient.

Bild 56: Anschlussprinzip für CANopen-Netzwerk

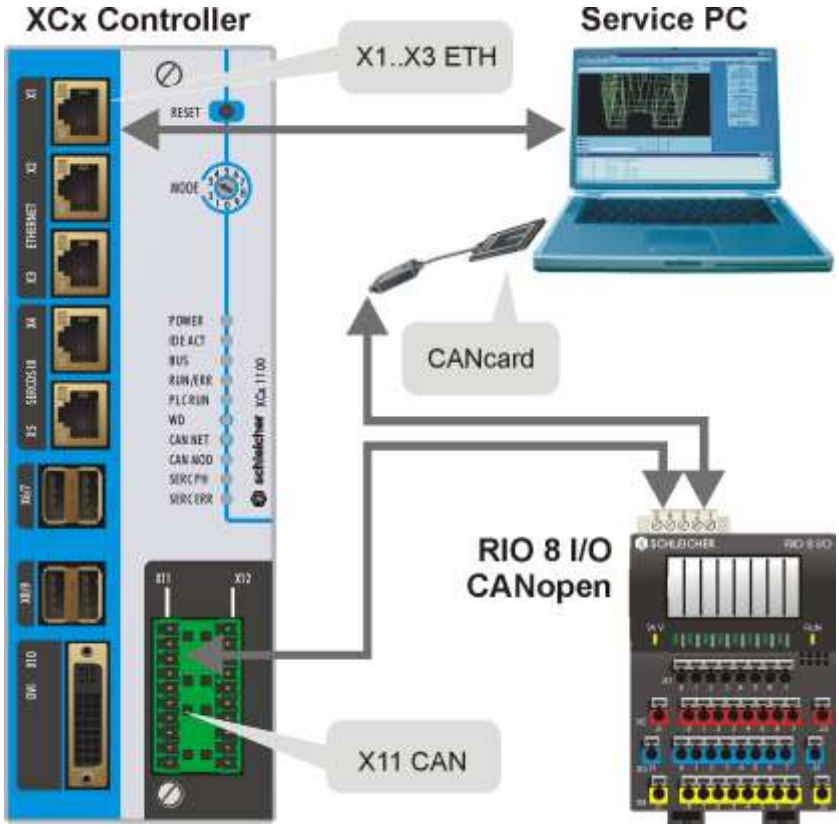
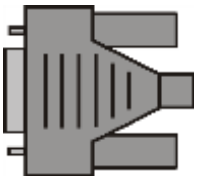
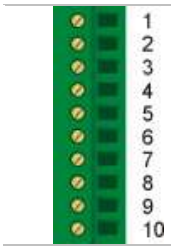


Tabelle 16: Pinbelegung der verwendeten Stecker





	1	**2**	**3**
	D-Sub, 9-polig, Buchse	Steckblock-klemme 10-pol.	Steckblock-klemme 5-pol.
	zur CANcard im Service-PC	X11, an der XCA Steuerung	am RIO 8 I/O CANopen
CAN	**Pin**	**Pin**	**Pin**
0 V		1 / 6	1
CAN_L	2	2 / 7	2**
Drain		3 / 8	3
CAN_H	7	4 / 9	4**
DC +24 V		5 / 10	5

* Die Pingruppen 1..5 und 6..10 sind parallel geschaltet.

** Zwischen Pin 2 und 4 am RIO 8 I/O CANopen muss ein Abschluss-widerstand von 120 Ohm geschaltet werden.

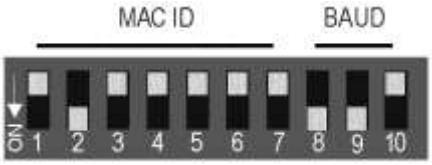
6.7.3 Einstellungen am I/O-Modul RIO 8 I/O CANopen

Am Kompaktmodul RIO 8 I/O CANopen die Knotennummer 2 und die Datenübertragungsrate 125 kBaud einstellen (Bild 57). Dazu den DIP-Schalter auf der Moduloberseite einstellen:

Tabelle 17: Einstellen von Knotennummer und Datenübertragungsrate

	Knotennummer MAC ID							**Datenübertragungsrate BAUD**		
Schalter	1	2	3	4	5	6	7	8	9	10
Stellung	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF

Bild 57: DIP-Schalter am Kompaktmodul RIO 8 I/O CANopen



6.7.4 Deklaration des I/O-Treibers für CANopen

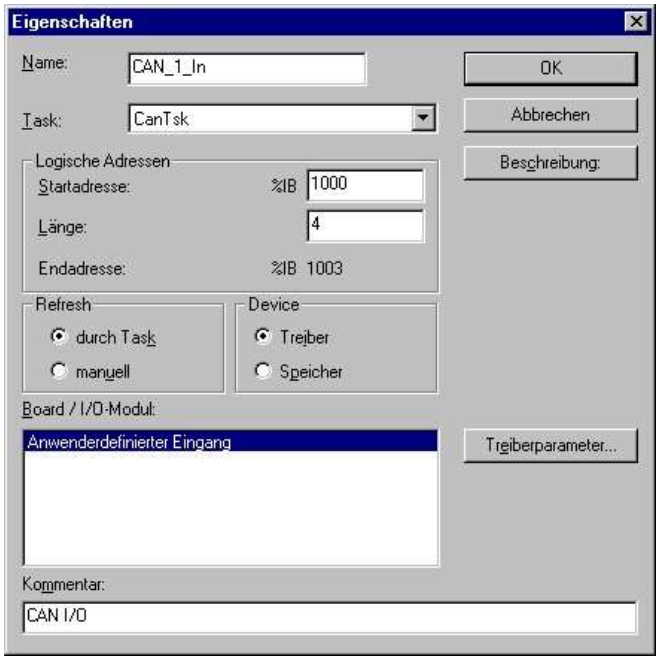
Ort und Prinzip der Deklaration des I/O-Treibers für das CANopen-Netzwerk sind dieselben wie für XUIO. Am Ende des Projektbaums befindet sich der Container *IO_Configuration*. Dort sind bereits die I/O-Konfigurationen *CAN_1_In* (unter INPUT) und *CAN_1_Out* (unter OUTPUT) erstellt (Bild 58).

Bild 58: CANopen-Konfiguration, Dialogfenster "I/O-Konfiguration"

Für dieses Beispiel müssen folgende Parameter eingetragen werden (Bild 59):

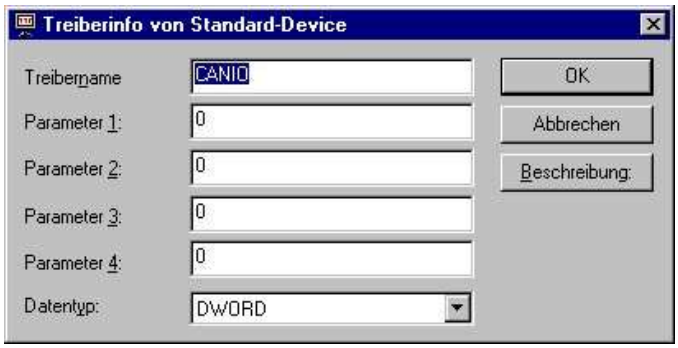
- Die **Task**, mit welcher der I/O Adressraum synchronisiert wird, muss *CanTsk* sein.
- Als **Startadresse** werden die logischen Adressen *IB1000* für *CAN_1_In* und *QB1000* für *CAN_1_Out* eingetragen.
- Im Parameter **Länge** werden so viele I/O Bytes deklariert, wie im CANopen-Netzwerk ausgetauscht werden sollen (hier 4, da minimal Doppelwortabstände angewendet werden).

Bild 59: CANopen-Konfiguration, Dialogfenster "Eigenschaften"



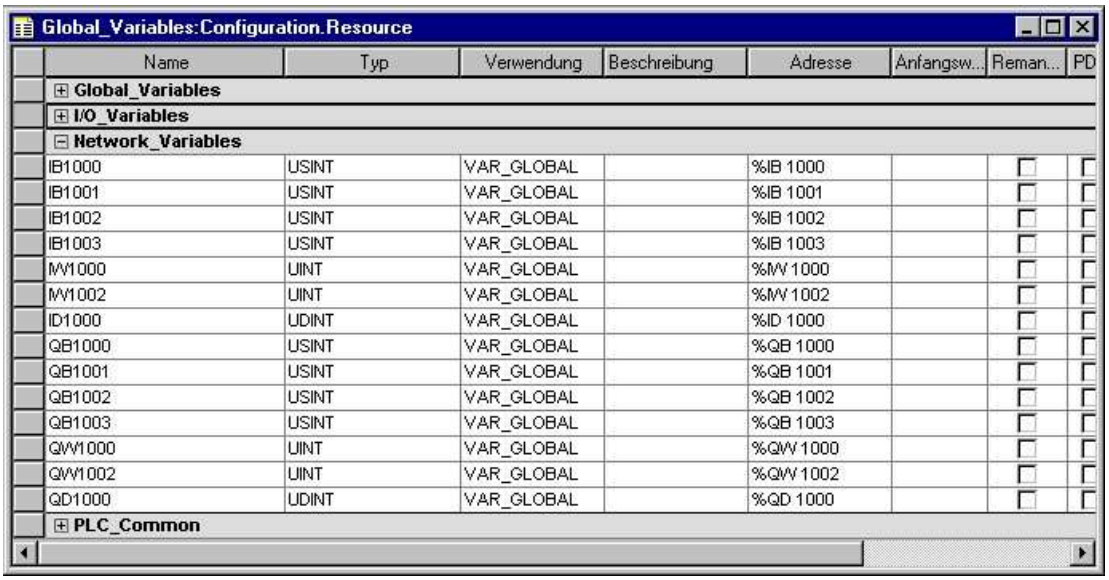
In den **Treiberparametern** muss der Treibername *CANIO* eingestellt sein, der Datentyp ist DWORD (Bild 60).

Bild 60: CANopen-Konfiguration, Treiberparameter einstellen



6.7.5 Deklaration von Netzwerkvariablen in MULTIPROG

Im Fenster des Projektbaums unter *Global_Variables* sind im Ordner *Network_Variables* die benötigten Variablen vordefiniert (Bild 61).



Global_Variables:Configuration.Resource

Name	Typ	Verwendung	Beschreibung	Adresse	Anfangsw...	Reman...	PD
⊞ Global_Variables							
⊞ I/O_Variables							
⊟ Network_Variables							
IB1000	USINT	VAR_GLOBAL		%IB 1000		☐	☐
IB1001	USINT	VAR_GLOBAL		%IB 1001		☐	☐
IB1002	USINT	VAR_GLOBAL		%IB 1002		☐	☐
IB1003	USINT	VAR_GLOBAL		%IB 1003		☐	☐
IW1000	UINT	VAR_GLOBAL		%IW 1000		☐	☐
IW1002	UINT	VAR_GLOBAL		%IW 1002		☐	☐
ID1000	UDINT	VAR_GLOBAL		%ID 1000		☐	☐
QB1000	USINT	VAR_GLOBAL		%QB 1000		☐	☐
QB1001	USINT	VAR_GLOBAL		%QB 1001		☐	☐
QB1002	USINT	VAR_GLOBAL		%QB 1002		☐	☐
QB1003	USINT	VAR_GLOBAL		%QB 1003		☐	☐
QW1000	UINT	VAR_GLOBAL		%QW 1000		☐	☐
QW1002	UINT	VAR_GLOBAL		%QW 1002		☐	☐
QD1000	UDINT	VAR_GLOBAL		%QD 1000		☐	☐
⊞ PLC_Common							

Bild 61: CANopen-Konfiguration, Arbeitsblatt "Network_Variables"

I/O-Bits werden im Arbeitsblatt *I/O_Variables* mit den Adressen *IX1000.0* und *QX1000.0* deklariert (Bild 62). (Im Beispiel wird *QX1000.7* benutzt, um das Ergebnis am RIO 8 I/O sichtbar zu machen.)

Global_Variables:Configuration.Resource

Name	Typ	Verwendung	Beschreibung	Adresse	An
⊞ Global_Variables					
⊟ I/O_Variables					
CAN_1_IX0	BOOL	VAR_GLOBAL	CANopen 1. Netzwerk Eingang 0	%IX 1000.0	
CAN_1_QX7	BOOL	VAR_GLOBAL	CANopen 1. Netzwerk Ausgang 7	%QX 1000.7	
XRIO_1_IX0	BOOL	VAR_GLOBAL	XRIO Modul 1 (RIO 16 I) Eingang 0	%IX 0.0	
XRIO_1_IX1	BOOL	VAR_GLOBAL	XRIO Modul 1 (RIO 16 I) Eingang 1	%IX 0.1	
XRIO_2_QX0	BOOL	VAR_GLOBAL	XRIO Modul 2 (RIO 16 O) Ausgang 0	%QX 0.0	

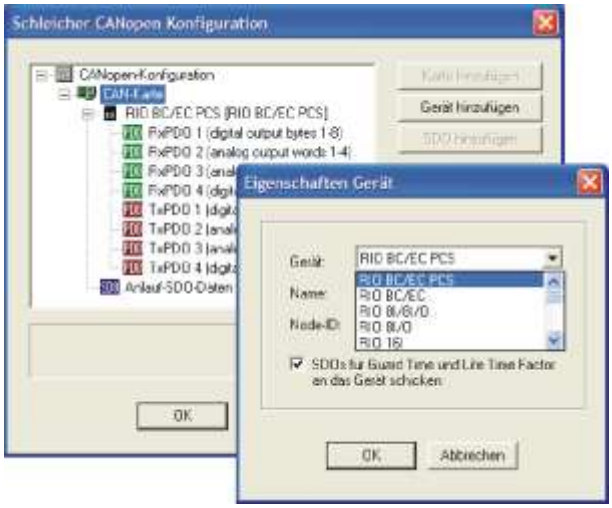
Bild 62: CANopen-Konfiguration, Arbeitsblatt "I/O_Variables"

6.7.6 CANopen-Konfiguration mit "Schleicher CANopen Konfiguration"

Für die Einrichtung einfacher CANopen-Netze steht die **Schleicher CANopen Konfiguration** zur Verfügung, die zusammen mit den MULTIPROG-AddOns installiert wird. Der Konfigurator ist ein reines Offline-Werkzeug, d.h. es besteht keine Kommunikation zur CAN-Karte der Steuerung. Das Werkzeug generiert Programmcode, der beim Start der SPS ausgeführt wird. Dabei werden eine Reihe von SDOs zur Steuerung übertragen, die das Netzwerk konfigurieren.

Bild 63:
Aufruf der "Schleicher CANopen Konfiguration" in MULTIPROG

Bild 64:
Schleicher CANopen Konfiguration



Im Konfigurationsfenster können (je nach Steuerung) eine oder mehrere CAN-Karten konfiguriert werden. CAN-Geräte wie Buskoppler oder Antriebe lassen sich einfach über eine Auswahlliste hinzufügen. Dabei werden automatisch die vom Gerät unterstützten Empfangs- und Sende-PDOs eingefügt. Anschließend wird die erstellte Konfiguration gespeichert und alle erforderlichen Änderungen am SPS-Projekt (inkl. Generierung des SPS-Programmcodes) vorgenommen.

Da der Konfigurator offline arbeitet, ist keine direkte Fehlerdiagnose mit dem Werkzeug selbst möglich. Es werden jedoch SPS-Programmcode sowie Variablen erzeugt, die es erlauben, die Konfigurierung des CANopen-Netzwerkes nach dem Start der SPS zu überprüfen. Die Werte dieser Variablen können im Online-Modus von MULTIPROG überprüft werden.

6.7.7 CANopen-Konfiguration mit "ProCANopen"

ProCANopen ist ein Konfigurationswerkzeug von der Fa. Vector Informatik GmbH

Komplexe CANopen-Netzwerke können mit dem **ProCANopen Konfigurationswerkzeug** konfiguriert werden. Dazu ist die Software "ProCANopen" erforderlich, die nicht im Lieferumfang von MULTIPROG enthalten ist. Zusätzlich wird eine CAN-Feldbuskarte im Service-PC benötigt, z.B. "CANcardY" (Seite 130).

Die Eigenschaften und Fähigkeiten der Komponenten sind in einer EDS-Datei (**E**lectronic **D**ata **S**heet) deklariert. Die EDS-Datei muss in das Unterverzeichnis von ProCANopen mit dem Namen EDS kopiert werden.

ProCANopen bildet die (abbildbaren) Objekte der Knoten ab. Beispielsweise werden Ausgangs-Bytes der RIO Module (die die Eingangs-Bits vom RIO repräsentieren) mit Eingangs-Bytes der XCx verknüpft. Darüber hinaus werden zusätzliche Informationen für den Feldbus konfiguriert:

- Welcher Knoten ist der "NMT manager"?
- Welcher Knoten ist der "Configuration manager"?
- "Guarding" und "Guarding time"
- "Sync time" und "Sync window length"

Nach der Netzwerkkonfiguration kann diese im Netzwerk gespeichert werden. Im Netzwerk speichern bedeutet: der Knoten, welcher als "Configuration manager" ausgewählt ist (meistens XCx), bekommt via CANopen die Information, wie das Netzwerk zu konfigurieren ist. Der "Configuration manager" speichert die Information (z.B. bei der XCx auf der Flashdisk), und nach dem Einschalten konfiguriert die XCx das Netzwerk.

Nachdem das Netzwerk konfiguriert ist, kann der "NMT manager" das CAN-Netzwerk starten. Netzwerk starten bedeutet: Status "operational" und Datenaustausch von PDOs (**P**rocess **D**ata **O**bjects).

6.7.8 Installation von ProCANopen

Hinweis

Für Anwender, welche bereits ProCANopen Version 2.1 und MULTIPROG 1.2 für die Schleicher-Steuerungen MicroLine und ProNumeric/ProSycon installiert haben:

- **Sie benötigen nur die Update-Version von ProCANopen.**
- **Überschreiben Sie nicht die bereits installierte Version!**
- **Installieren Sie ProCANopen V3.2 in einem neuen Pfad auf der Harddisk z.B. \ProCANopen3.**
- **Wenn Sie eine CANCardX im Einsatz haben, ist es manchmal erforderlich, ein Update der Firmware und der Options auf der Karte vorzunehmen: Bitte notieren Sie die Seriennummer der Karte und setzen Sie sich mit Ihrem lokalen Händler in Verbindung.**
- **In Abhängigkeit von Ihrem PC-Betriebssystem müssen Sie verschiedene Treiber installieren. Einige neue Treiber sind nicht kompatibel mit der älteren ProCANopen-Version 2.1! Dies bedeutet, ProCANopen Version 2.1 mit dem neuen Treiber V3.x funktioniert nicht mehr online mit dem Feldbus!**
- **Mit ProCANopen V2.1 projektierte Konfigurationsdateien können mit ProCANopen V3.2 weiter benutzt werden.**

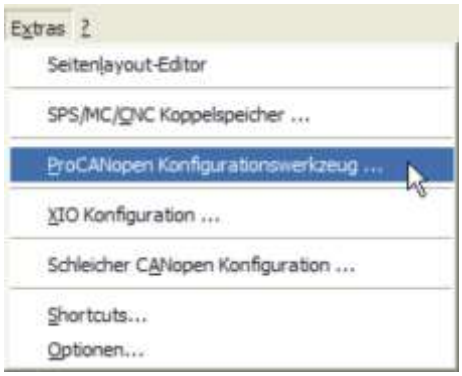
Um ProCANopen zu installieren, folgen Sie bitte der Dokumentation, die mit der Software und der CAN-Karte geliefert wurde. Sie müssen die Treiber und die ProCANopen-Software in zwei Schritten installieren:

- Kopieren sie die aktuellen EDS-Dateien für den Steuerungstyp von der Service-CD in das Verzeichnis *\...\ProCANopen\EDS*.
- Wenn die Dialogsprache Englisch gewünscht ist, ändern Sie in der Datei *\....\ProCANopen\EXE\VECTOR.INI* die Zeile "language=0049" in "language=001".

6.7.9 Einbindung von ProCANopen in MULTIPROG

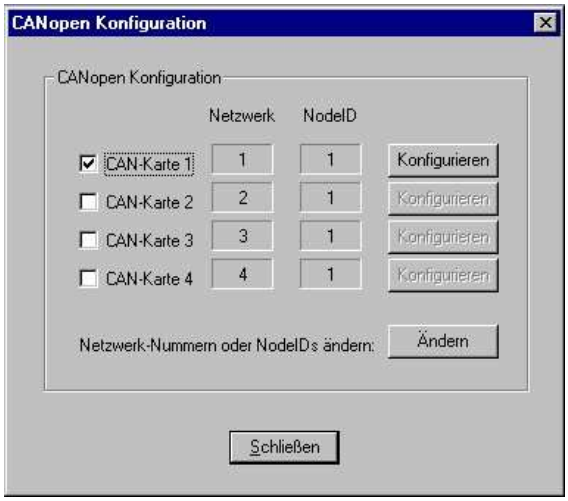
Die Installation der AddOns bereitet MULTIPROG so vor, dass ProCANopen direkt von MULTIPROG aus gestartet werden kann. Dazu im Menü *Extras / ProCANopen* Konfigurationswerkzeug wählen (Bild 65).

Bild 65:
Aufruf von "ProCANopen Konfigurationswerkzeug" in MULTIPROG



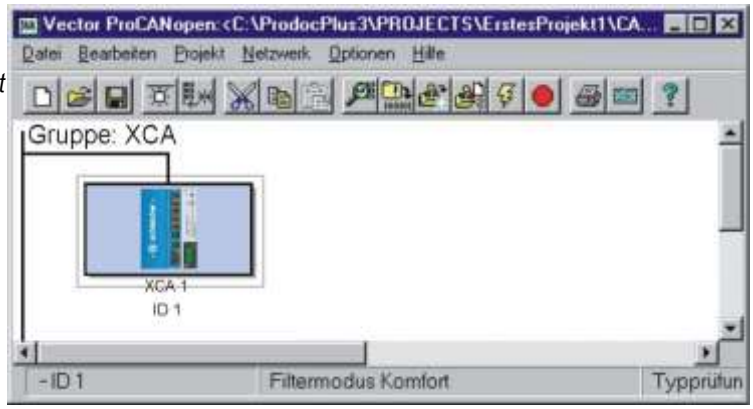
Die aktive CAN-Karte der XCx anwählen. Im Beispiel wird nur die eine standardmäßig vorhandene Karte benutzt. Als Knotennummer (NodeID) kann 1 beibehalten werden. Zum Start von ProCANopen auf den Button *Konfigurieren* klicken (Bild 66).

Bild 66:
Auswahl der CAN-Karte und Starten von ProCANopen



ProCANopen startet direkt mit dem richtigen CANopen-Projekt (Bild 67).

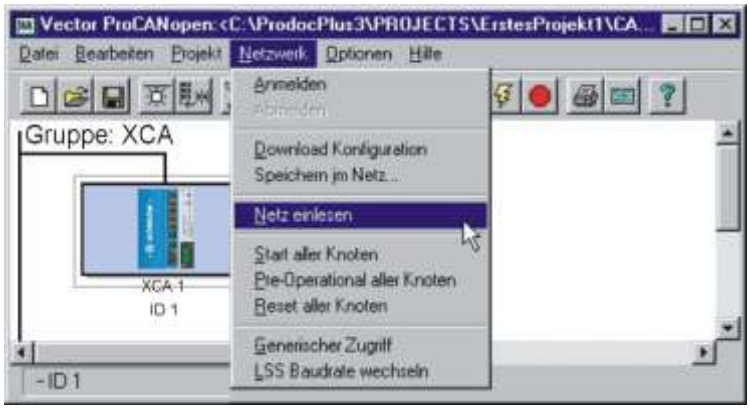
Bild 67:
ProCANopen mit aktuellem CANopen-Projekt



6.7.10 Erste Verbindungen mit ProCANopen

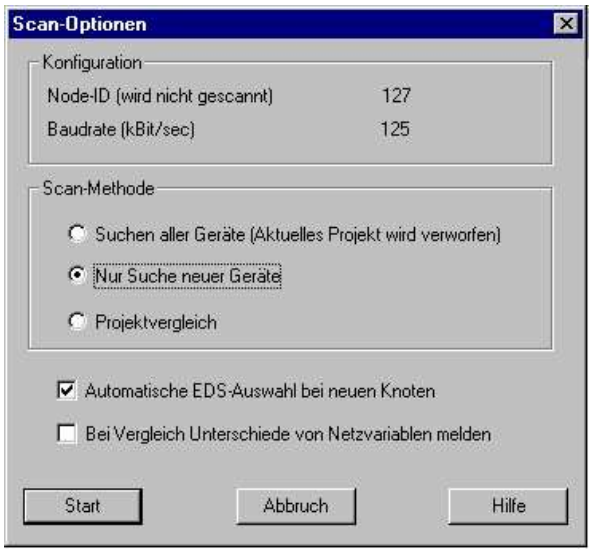
Zuerst muss das Netzwerk eingelesen werden (Bild 68).

Bild 68:
Einlesen des Netzes mit ProCANopen



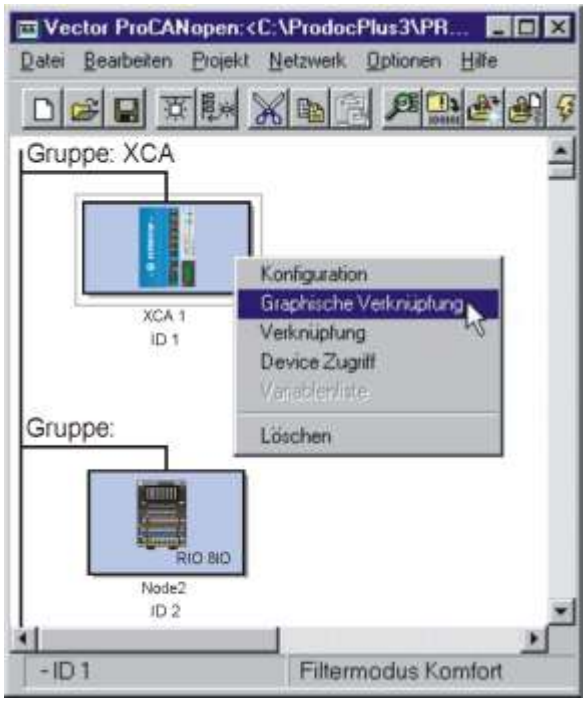
Da das Netzwerk mit "Knoten 1 XCA" bereits vorkonfiguriert ist, muss das Einlesen mit der Scan-Option *Nur Suche neuer Geräte* ausgeführt werden (Bild 69).

Bild 69:
Einstellen der Scan-Optionen



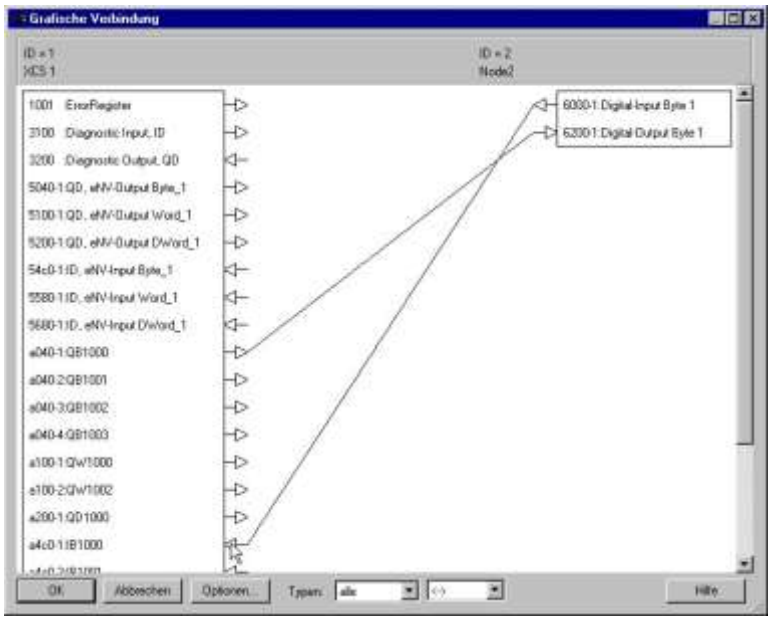
Dann können die Verknüpfungen der Netzwerk-Knoten projektiert werden. Dazu mit der rechten Maustaste auf die XCx klicken, im Kontextmenü des Knotens *Graphische Verknüpfung* auswählen und dann auf den Knoten klicken, mit dem die Verbindung hergestellt werden soll (im Beispiel Knoten 2 RIO 8 I/O) (Bild 70).

Bild 70:
Graphische Verknüpfung der Netzwerk-Knoten



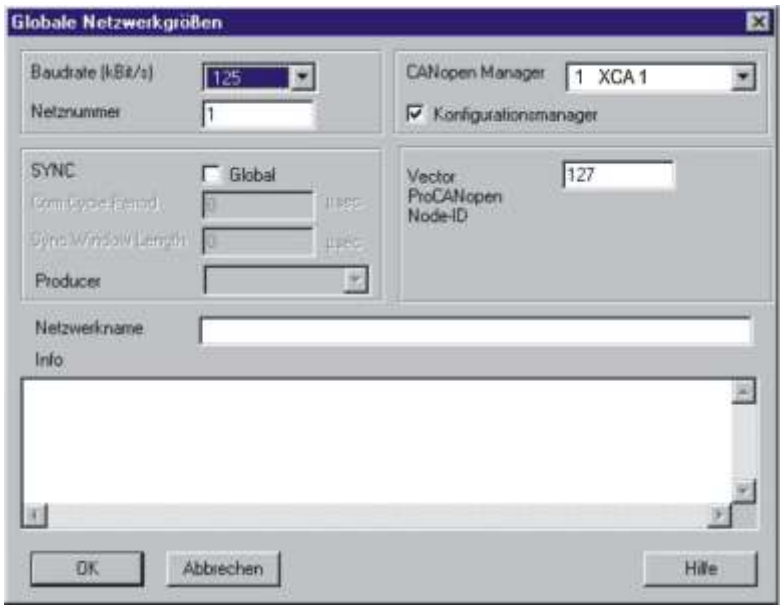
Das nächste Bild zeigt beispielhaft die Kommunikationsbeziehungen zwischen der Steuerung und dem I/O-Modul (Bild 71).

Bild 71:
Kommunikations beziehung zwischen Steuerung und I/O-Modul



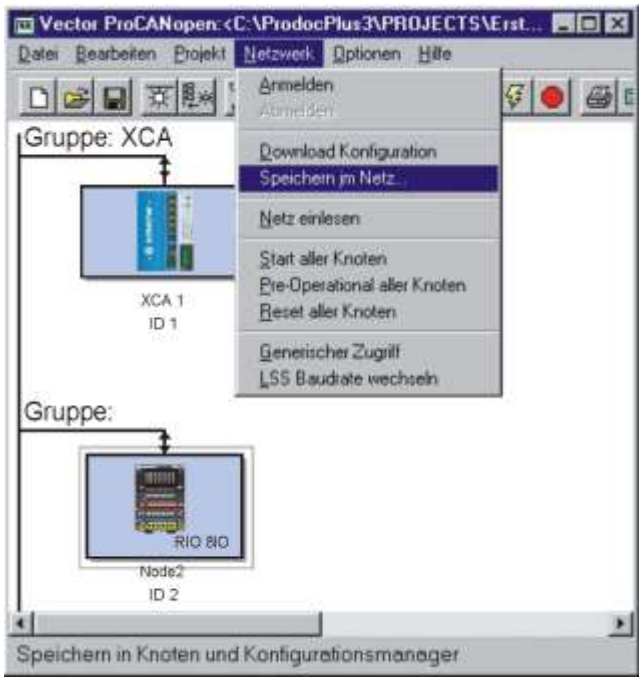
In *Projekt / Globale Konfiguration* den CANopen-Manager "XCA 1" als Konfigurationsmanager auswählen (Bild 72).

Bild 72:
Auswahl des Konfigurationsmanagers



Mit *Speichern im Netz* wird die CAN-Konfiguration in den Konfigurationsmanager gespeichert (Bild 73). Die XCx speichert die Daten in den Compact Flash, und nach dem Einschalten wird das Netzwerk gebootet.

Bild 73:
Speichern der CAN-Konfiguration in den Konfigurationsmanager

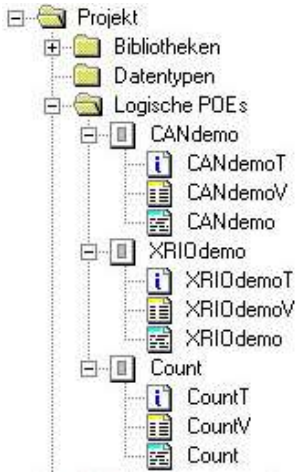


Hinweis

Weitere Hinweise zur CAN-Konfiguration finden Sie in der Betriebsanleitung "Inbetriebnahmehinweise für Feldbussysteme" (siehe 10).

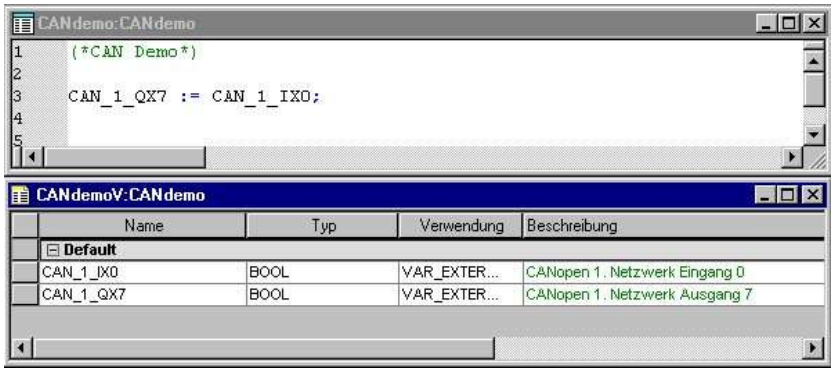
Zum Test der Netzwerkverbindung muss eine neue POE (hier *CANdemo*) erstellt und in der *CanTsk* instanziiert werden (Bild 74).

Bild 74:
Test der Netzwerk-verbindung mit POE "CANdemo"



Die POE *CANdemo* mit dem dazugehörigen Variablen-Arbeitsblatt (Bild 75):

Bild 75:
POE CANdemo mit dazugehörigem Variablen-Arbeitsblatt



Wird an den Eingang 0 vom RIO 8 I/O CANopen 24 V angelegt, wird der Ausgang 7 auf 1 gesetzt.

6.8 Der Webserver

6.8.1 Allgemeine Funktionen und Konzept

- Der Hauptvorteil der Webserver-Technologie ist die Ablage der kompletten Visualisierungsapplikation (HTML, JavaScript) auf der Steuerung. Eine zusätzliche Konfiguration in einem Bediengerät entfällt.
- Der Webserver ist im Betriebssystem der XCx integriert.
- Der Browser ist der "thin client" für die Datenvisualisierung.
- Andere Visualisierungssysteme müssen auf jedem Bediengerät gesondert installiert werden. Dieses Konzept nennt man "fat client".

6.8.2 Schleicher-spezifisches Applet

Normalerweise ist die Web-Technologie ein Herunterladen in einer Richtung zum Browser, die Web -Seite selbst ist dynamisch (Animation Gifs oder Flash Files). Ein zyklischer Parameterrefresh ist nicht möglich.

Schleicher liefert ein spezielles Java-Applet, das einen Datenaustausch bidirektional zwischen Browser und Steuerung ermöglicht. Dieses Applet unterstützt Funktionen, die von der HTML/Java script language aufgerufen werden können. Diese Funktionen ermöglichen der Applikation, einzelne oder mehrere Variablenwerte der SPS zu schreiben.

6.8.3 Deklaration von Variablen zur Visualisierung

Die Variablen, die visualisiert werden sollen, sind in MULTIPROG mit der Checkbox *PDD* zu markieren (PDD = **P**rocess **D**ata **D**irectory).

Bild 76:
Deklaration von PDD-Variablen in MULTIPROG

Diese Variablen werden in der XCx in einer internen Liste aktualisiert. Der Webserver kann die Variablen dieser Liste lesen und schreiben.

6.8.4 Applikationsbeispiel

Die XCx wird mit einer Standard-Browser-Applikation ausgeliefert. Diese Applikation ermöglicht, PDD-markierte Variablen zu lesen und zu schreiben. Zusätzlich wird ein Statusüberblick gegeben.

6.8.5 Browser / Komponenten

- Alle Standard-PCs mit Ethernet sind möglich.
- MS Internet Explorer V5 oder höher sind erforderlich.
- Einige Terminals mit Windows CE sind möglich, wenn der Browser die Anforderungen wie Java Script 1.5, Java 2, HTTP1.1 erfüllt.

6.9 Allgemeine Hinweise zur Inbetriebnahme

Grundinitialisierung

Um bei fehlerhaften SPS-Projekten, Q-Parametern bzw. ungültigen Daten im remanenten Datenspeicher (z.B. bei Ausfall der Pufferbatterie) den Betrieb des Echtzeitbetriebssystems sicherzustellen, kann die Steuerung mit Hilfe der sogenannten Grundinitialisierung in einem abgesicherten Modus starten. Die Grundinitialisierung bewirkt ein Zurücksetzen des SRAMs.

Hiermit besteht die Möglichkeit einer Diagnose der Steuerungsdaten.

Durchführen der Grundinitialisierung

Bitte beachten: Für alle Übergänge gilt jeweils ein Timeout von 4 s.

- Betriebsartenschalter in Position "0".
- Einschalten der Steuerung (bzw. "START XCA 11xx" im MFA-Tool).
- Das gleichzeitige, gelbe Blinken aller LEDs signalisiert den Start der "GrundInit"-Erkennung.
- Betriebsartenschalter in Position "9", jetzt blinken nur die LEDs *RUN/ERR* und *PLC RUN* rot weiter.
- Betriebsartenschalter in Position "0" zurückschalten, alle LEDs blinken jetzt rot und zeigen die Durchführung der Grundinitialisierung an.

Nach der Grundinitialisierung wird die Steuerung im "Abgesicherten Modus" gestartet. Im "Abgesicherten Modus laufen weder SPS noch CNC, der Zugriff auf die Steuerung über FTP ist aber möglich.

Hinweis

Wird der Drehschalter nach dem Einschalten der Steuerung in Stellung "0" belassen, wird das SPS-Bootprojekt nicht geladen und die SPS startet nicht.

7 Bedienung

7.1 Multi Function Application MFA

MFA (Multi Function Application) bildet die Grundlage für die Bedienung der Steuerung. Mit der MFA kann das Startverhalten eingestellt und die Steuerung sowie die SPS gestartet und angehalten werden.

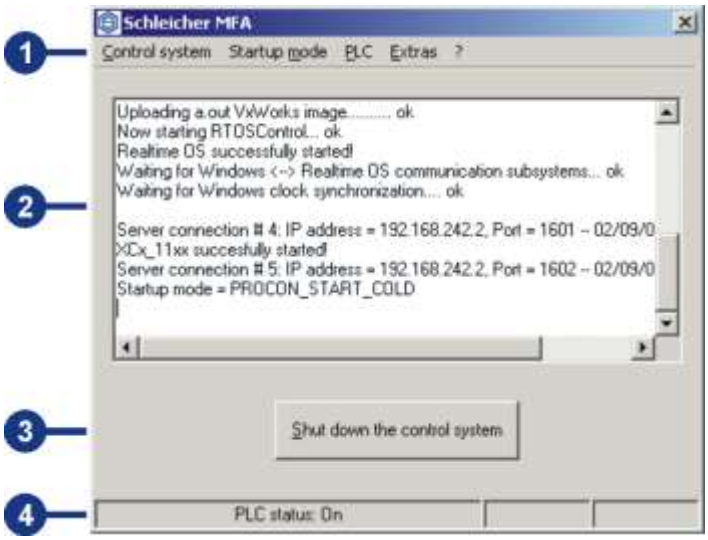
7.1.1 Start der MFA

MFA wird beim Hochlaufen der Steuerung automatisch gestartet und in den System-Tray der Taskleiste eingetragen (Bild 77). Von dort kann sie durch Doppelklick auf das blaue Schleicher-Logo aktiviert werden.

Bild 77:
Taskleiste mit Schleicher-Logo

7.1.2 Fensteraufbau der MFA

Bild 78:
Fensteraufbau der MFA

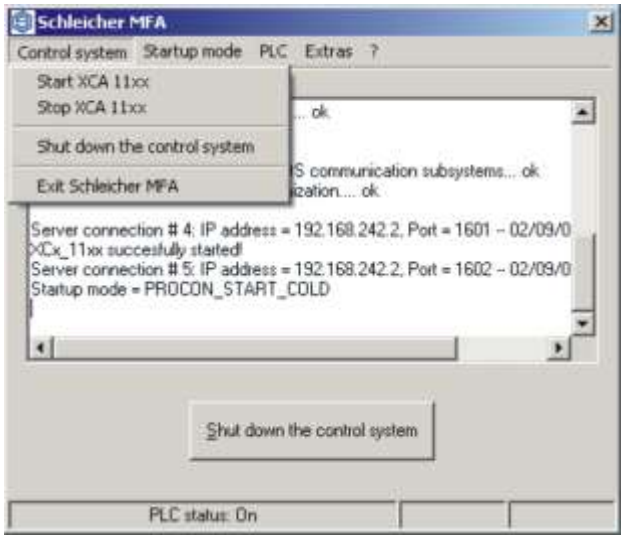


1	**Menüleiste** Erklärung siehe unter "Funktionen der MFA".
2	**Meldebereich** Mit Informationen zum Speicher, Echtzeitbetriebssystem, und Steuerungssoftware.
3	**Buttonbereich**
4	**Statusbereich** Informationen über den Zustand der SPS.

7.1.3 Funktionen der MFA

Menü "Control System"

Bild 79:
MFA, Menü
"Control System"

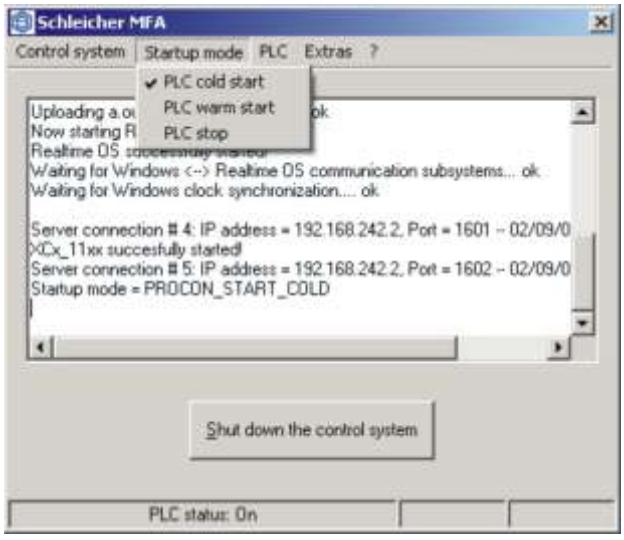


Mit den Menüpunkten *Start ... / Stop ...* werden das Echtzeitbetriebssystem und die Steuerungssoftware gestartet und beendet.

Der Menüpunkt *Shut down the control system* entspricht in seiner Funktion dem Button <Shut down the control system>. Die Steuerungssoftware einschließlich PC-Betriebssystem wird heruntergefahren und die Steuerung ausgeschaltet.

Menü "Startup Mode"

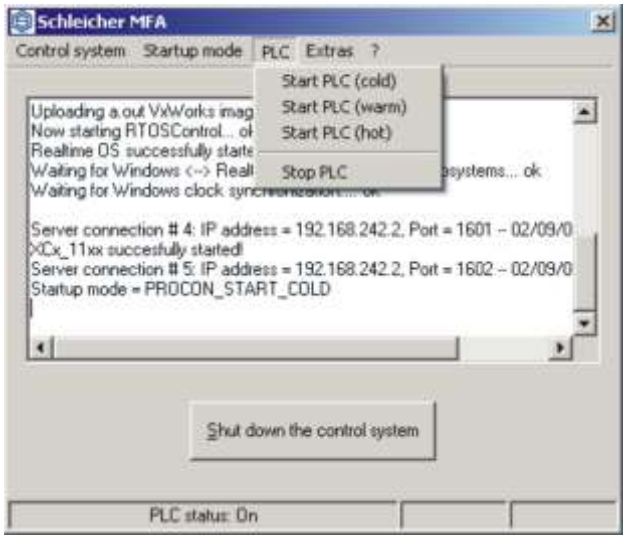
Bild 80:
MFA, Menü
"Startup Mode"


Schleicher MFA
Control system
Startup mode
PLC
Extras
?
PLC cold start
PLC warm start
PLC stop
Waiting for Windows <--> Realtime OS communication subsystems... ok
Waiting for Windows clock synchronization.... ok
Server connection # 4: IP address = 192.168.242.2, Port = 1601 -- 02/09/0
XCx_11xx succesfully started!
Server connection # 5: IP address = 192.168.242.2, Port = 1602 -- 02/09/0
Startup mode = PROCON_START_COLD
Shut down the control system
PLC status: On

Im Menü *Startup Mode* kann das Startverhalten der SPS nach dem Hochlaufen der Steuerung eingestellt werden. Das Startverhalten wird im Kapitel "Die SPS" weitergehend beschrieben (Seite 90).

Menü "PLC"

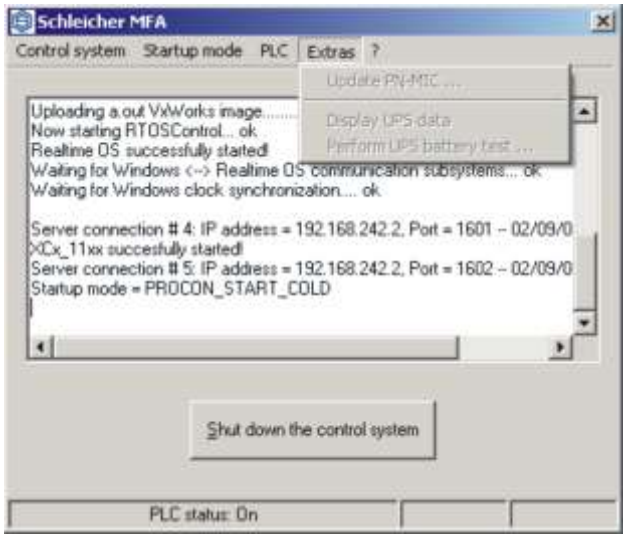
Bild 81:
MFA, Menü "PLC"



Im Menü *PLC* wird nur die SPS gestartet und angehalten. Das Startverhalten wird im Kapitel "Die SPS" weitergehend beschrieben (Seite 90).

Menü "Extras"

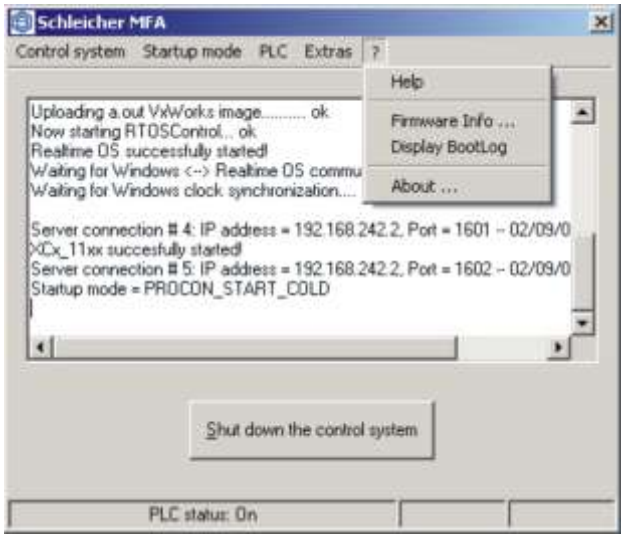
Bild 82:
MFA, Menü "Extras"


Schleicher MFA
Control system
Startup mode
PLC
Extras
?
Update PN-MIC ...
Display UPS data
Perform UPS battery test ...
Uploading a.out VxWorks image.....
Now starting RTOSControl... ok
Realtime OS successfully started!
Waiting for Windows <--> Realtime OS communication subsystems... ok
Waiting for Windows clock synchronization.... ok
Server connection # 4: IP address = 192.168.242.2, Port = 1601 -- 02/09/0
XCx_11xx successfully started!
Server connection # 5: IP address = 192.168.242.2, Port = 1602 -- 02/09/0
Startup mode = PROCON_START_COLD
Shut down the control system
PLC status: On

Im Menü *Extras* kann der Zustand einer eventuell über USB angeschlossenen UPS (USV / Unterbrechungsfreie Stromversorgung) angezeigt und getestet werden. Unterstützt werden aktuell UPS der Fa. Eaton Powerware (z.B. Powerware 5115 500 VA).

Menü "?"

Bild 83:
MFA, Menü "?"



Über das Menü *?* kann die MFA-Dokumentation aufgerufen werden. Die Hilfedatei enthält Informationen zu Programmeinstellungen und Kommadozeilenparametern sowie Programmierbeispiele.

Mit dem Menüeintrag *Display BootLog* wird die BootLog-Datei des Echtzeitbetriebssystems VxWorks angezeigt, die beim Steuerungsanlauf generiert wird.

Speicherort und Name der Datei: *SCHLEICHER/Os/Log/bootlog.txt*

7.1.4 Die Log-Datei der MFA

MFA erzeugt eine Log-Datei mit Fehlermeldungen und Bootprotokoll des Echtzeitbetriebssystemes VxWorks. Diese Dateien werden mit jedem Neustart der Steuerung erzeugt. Ist die Steuerung im Dauerbetrieb, dann wird die Datei LogBook_xx.txt alle 24 Stunden (Defaulteinstellung) gespeichert. In dem Log-Ordner sind max. 10 Dateien vom Typ BootLog_xx.txt und LogBook_xx.txt abgelegt. Die aktuelle Log-Datei kann in der Datei FileList.ini ermittelt werden, der Dateiname _xx entspricht dem Inhalt von Count (Beispiel Count =5)

Speicherort und Name der Datei:

SCHLEICHER/Os/Log/ BootLog_05.txt

SCHLEICHER/Os/Log/ LogBook_05.txt

Hinweis

Die Log-Dateien sollen vor allem für die Diagnoseunterstützung durch den Steuerungshersteller eingesetzt werden.

7.1.5 Kommunikation mit anderen Applikationen

MFA besitzt eine Schnittstelle zur Kommunikation mit anderen Windows-Applikationen (z.B. einem Visualisierungsprogramm). Die Kommunikation von MFA mit einer anderen Applikation erfolgt über eine registrierte Windows-Nachricht. Durch den Aufruf der Windows-API-Funktion *RegisterWindowMessage()* mit dem String "Schleicher MFA" als Parameter erhält eine Applikation den numerischen Wert der Windows-Nachricht, die zur Kommunikation mit MFA vorgesehen ist.

Weitere Informationen hierzu sowie ein Programmierbeispiel finden Sie in der MFA-Hilfe (Menü "?").

7.2 Schleicher-Dialog

Der Schleicher-Dialog stellt alle Dialoge zum Betrieb der CNC und SPS zur Verfügung. Der Schleicher-Dialog ist auf der Steuerung fest installiert und startet automatisch nach dem Steuerungsanlauf.

7.2.1 Aufbau der Bedienoberfläche

Bild 84: Schleicher-Dialog, Startfenster

Tabelle 18: Schleicher-Dialog, Aufteilung der Bedienoberfläche

Bereich	Bedeutung
1	Status- und Meldebereich
2	Arbeitsbereich für Einstellungen und Informationen
3	Hinweisbereich
4	Softkeys mit Funktionshinweis

Bild 85: Schleicher-Dialog, Status- und Meldebereich

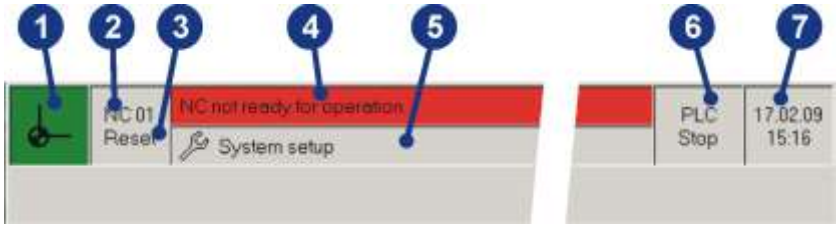


Tabelle 19: Schleicher-Dialog, Status- und Meldebereich

Bereich	Bedeutung
1	Aktuelle Betriebsart anzeigen
2	Angewähltes NC-Teilsystem
3	Aktueller NC-Zustand
4	Meldungfenster
5	Aktuelle Position im Steuerungsmenü
6	SPS-Status
7	Datum und Uhrzeit

7.2.2 Schleicher-Dialog SPS/CNC

Steuerungsmenü und Bedienbereiche der XCx

Die oberste Ebene des Steuerungsmenüs besteht aus Bedienbereichen, die sich an den für die Maschine wichtigen Tätigkeiten orientieren (Handbetrieb, Automatik, Programmieren, etc.). Sie werden mit der Tastenkombination <Strg + Funktionstaste> aufgerufen.

Über die untergeordnete Softkey-Ebene (Funktionstasten F1..F8) werden zugehörigen Optionen aufgerufen.

Der Softkey *F1* wird immer zum Aufruf der Hilfeseiten verwendet. Die Hilfeseiten enthalten weitergehende Informationen zum Inhalt der tieferliegenden Bedienebenen.

Handbetrieb <Strg+F1>

F1 Hilfe	F2 Referen-zieren	F3 Verfahren	F4	F5	F6	F7 Teilsystem	F8
	Achsen	Jog					
		Schrittmaß					
		Zielwert-vorgabe					
		Handrad					
		Achse					
		Eilgang					
		Nullsetzen					
		Override					

Automatik <Strg+F2>

F1 Hilfe	F2 Programm	F3	F4	F5	F6	F7 Teilsystem	F8
	Aktivieren						
	MDI						
	Einzelsatz						
	Blocksatz						
	Satzfolge						
	Eilgang						
	Override						

Programmieren <Strg+F3>

F1 Hilfe	F2 CNC-Programme	F3 R-Parameter	F4 Werkzeug-daten	F5 Nullpunkt-verschie-bungen	F6 Koordinaten-systeme	F7	F8
	Programm editieren	Wert ändern	Wert ändern	Wert ändern	Wert ändern		
	Neues Programm	Editieren					
	Programm aktivieren	Neu					
	Programm kopieren	Löschen					
	Programm löschen						
	Programm schützen						
	Neues Projekt						
	Projekt aktivieren						
	Projekt kopieren						
	Projekt löschen						
	Projekt schützen						
	Ansicht						
	Editieren						
	Neu						
	Übertragen						
	Aktua-lisieren						
	Löschen						
	Verzeichnis						

Zugangsberechtigung <Strg+F4> (gilt auch für XCS)

F1 Hilfe	F2	F3	F4	F5	F6	F7	F8

Fremdsoftware starten <Strg+F5> (gilt auch für XCS)

F1 Hilfe	F2	F3	F4	F5	F6	F7	F8
	Start Applikation 1	Start Applikation 2	Start Applikation 3	Start Applikation 4	Start Applikation 5	Start Applikation 6	Start Applikation 7

Inbetriebnahme <Strg+F6> (Softkeyebene 1)

				(Gilt auch für XCS)			
F1 Hilfe	**F2 Grundein-stellungen**	**F3 CNC-System**	**F4 Antriebs-konfiguration**	**F5 Boot-Ein-stellungen**	**F6 OPC-Variablen**	**F7 Info**	**F8 Weiter >>**
	Editieren	Editieren	Editieren	Editieren	Wert ändern		
		Anzeige-modus	Antriebs-parameter	CAN-Einstellungen	Editieren		
		Achs-zuordnung	DriveTop		Neu		
					Löschen		

Inbetriebnahme <Strg+F6> (Softkeyebene 2)

(Gilt auch für XCS)				(Gilt auch für XCS)			
F1 Hilfe	**F2 Programm-einstellungen**	**F3 SPS/CNC-Optionen**	**F4 Protokol-lierung**	**F5 Uhr stellen**	**F6 Verbinden**	**F7 Info**	**F8 << Zurück**
	Eigen-schaften	Optionen freigeben	Über-nehmen				
	Ressource hinzufügen		System-parameter hinzufügen				
	Ressource löschen		System-parameter löschen				
	Ressource nach oben		Antriebs-parameter hinzufügen				
	Ressource nach unten		Antriebs-parameter löschen				

7.2.3 Aufruf von Activ-Error-Buffer und Log-Book

Die Fehlermeldungen im Active-Error-Buffer und Log-Book sind auf jeder Bedienebene über die Tastenkombination <Strg+?> aufrufbar.

Fehler <Strg+?>							
F1 **Hilfe**	**F2** **Logbuch**	**F3** **Boot-Protokoll**	**F4** **Problem-report**	**F5**	**F6**	**F7**	**F8**
	Öffnen						
	Speichern						
	Drucken						
	Aktuali-sieren						
	Löschen						

8 Die SPS

- Betriebssytem: ProConOS
- Programmierung: MULTIPROG nach IEC 61131-3
- Kommunikation mit der CNC über Koppelspeicher

8.1 Programmierung

Die Programmierung der XCx erfolgt mit der Programmiersoftware MULTIPROG nach IEC 61131-3 auf einem PC.

Wichtig!

Die Programmiersoftware besteht aus der Software MULTIPROG und den AddOns für MULTIPROG von Schleicher.

Das Programmiersystem mit Programmieranleitung ist als Zubehör zu beziehen (Seite 130).

Die SPS wird mit dem fertig konfigurierten Projekt ausgeliefert, auf dessen Grundlage die Programmierung der SPS begonnen werden kann (siehe " Erste Schritte mit MULTIPROG", Seite 42).

8.2 SPS-Betriebszustände und Startverhalten

8.2.1 Betriebszustände

Betriebszustand	Beschreibung
EIN	• Es ist kein Programm geladen
STOP	• Programm ist geladen • Anwendertasks sind inaktiv • Eingänge des Prozessabbildspeichers werden nicht aktualisiert • Ausgangssignale werden nicht an die Ein- und Ausgänge übermittelt
BETRIEB	• Programmausführung ist aktiviert • Anwendertasks sind aktiv • Eingänge des Prozessabbildspeichers werden gemäß der I/O-Konfiguration aktualisiert • Ausgänge des Prozessabbildspeichers werden gemäß der I/O-Konfiguration und der Programmausführung aktualisiert
HALT	• Programmausführung wird an einem Haltepunkt angehalten • Anwendertasks sind inaktiv • Eingänge des Prozessabbildspeichers werden nicht aktualisiert • Ausgänge des Prozessabbildspeichers werden nicht aktualisiert

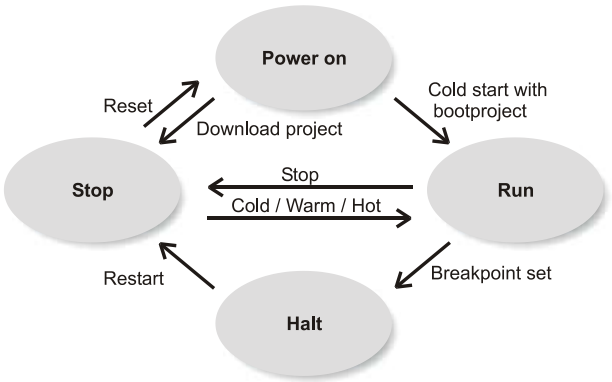
Tabelle 20: SPS-Betriebszustände

Der aktuelle Zustand der SPS wird in MULTIPROG im Projekt-Kontrolldialog in der Zeile *Status* angezeigt. Wenn hinter dem aktuellen Zustand im Kontrolldialog 'Debug' angezeigt wird, bedeutet das, dass Haltepunkte gesetzt oder Variablen geforct wurden.

8.2.2 Wechseln der Betriebszustände mit MULTIPROG

Über die graphische Benutzeroberfläche von MULTIPROG kann gesteuert werden, wann die Programmausführung auf der SPS gestartet und gestoppt wird. Die Schaltflächen für Wechsel, die im aktuellen Betriebszustand nicht möglich sind, sind im Projekt-Kontrolldialog abgeblendet (Bild 86).

Bild 86: MULTIPROG, Wechseln der Betriebszustände



Starten der Programmausführung

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Stop → Betrieb	Cold	• es erfolgt ein Kaltstart • alle Daten werden initialisiert • SPG 1 wird aufgerufen • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert
Stop → Betrieb	Warm	• es erfolgt ein Warmstart • nur nicht-gepufferte Daten werden initialisiert • SPG 0 wird aufgerufen • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert
Stop → Betrieb	Hot	• es erfolgt ein Heißstart • es werden keine Daten initialisiert • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert • nicht verfügbar, wenn Sie die Programmausführung zum ersten Mal nach dem Senden starten

Tabelle 21: MULTIPROG, Starten der Programmausführung

Stoppen der Programmausführung

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Betrieb → Stop		• alle Anwendertasks werden deaktiviert, wenn ihr Arbeitszyklus beendet ist • SPG 2 wird aufgerufen • die Ausgänge des Prozessabbildspeichers werden geschrieben • die Programmausführung wird gestoppt • die physikalischen Ausgänge werden auf Null oder Vorzugsabschaltlage gesetzt

Tabelle 22: MULTIPROG, Stoppen der Programmausführung

Allgemeines Reset

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Stop → Ein		• das Projekt wird gelöscht • es erfolgt ein allgemeines Reset

Tabelle 23: MULTIPROG, Allgemeines Reset

8.2.3 Startverhalten der SPS nach dem Einschalten der Versorgungsspannung

Das SPS-Startverhalten wird mit dem Betriebsartenschalter eingestellt. Es können folgende Varianten ausgewählt werden:

Tabelle 24: Betriebsartenschalter

Stellung / Bezeichnung	Bedeutung
0	Grundinitialisierung / Diagnose (Start des Echtzeitbetriebs-systems im abgesicherten Modus und Rücksetzen des remanenten Datenspeichers, (S. 79)
1 / Prog	Betriebsart Programmierung (SPS-Stopp)
2 / Warm (auch 4..9)	Warmstart der SPS nach IEC 61131-3 (Defaultstellung)
3 / Cold	Kaltstart der SPS nach IEC 61131-3 (Reinitialisierung der Retainvariablen)

8.3 Systemvariablen

Systemvariablen informieren über den Systemzustand, wie z.B. über geforcte Variablen, Leistungsfähigkeit der CPU, etc. Diese Variablen haben feste Speicheradressen und können vom SPS-Programm verwendet werden, um die entsprechenden Informationen zu erhalten.

Alle Systemvariablen in der folgenden Tabelle sind bereits im Bereich *Global_Variables* des Arbeitsblattes *Global_Variables* deklariert.

Name	Datentyp	Log. Adr. (Byte)	Log. Adr. (Bit)	Beschreibung
PLCMODE_ON	BOOL	0	0	TRUE := aktueller SPS-Zustand ist EIN
PLCMODE_RUN	BOOL	0	1	TRUE := aktueller SPS-Zustand ist BETRIEB
PLCMODE_STOP	BOOL	0	2	TRUE := aktueller SPS-Zustand ist STOP
PLCMODE_HALT	BOOL	0	3	TRUE := aktueller SPS-Zustand ist HALT
PLCDEBUG_BPSET	BOOL	1	4	TRUE := ein oder mehrere Haltepunkte sind gesetzt
PLCDEBUG_FORCE	BOOL	2	0	TRUE := eine oder mehrere Variablen sind geforct
PLCDEBUG_POWERFLOW	BOOL	2	3	TRUE := Durchlaufkontrolle ist aktiv
PLC_TICKS_PER_SEC	INT	44	-	Anzahl der Systemticks pro Sekunde, die von der SPS als Systemzeitbasis verwendet werden. Dieser Wert bestimmt die Zeitauflösung der SPS für Funktionsbausteine für Zeitverzögerung, wie TON, TOF oder TP und die kürzeste Zykluszeit für die DEFAULT-Task und zyklische Tasks.
PLC_SYS_TICK_CNT	DINT	52	-	Anzahl der gezählten SPS Systemticks

Tabelle 25: Systemvariablen

Zusätzlich zu diesen Systemvariablen sind weitere Variablen definiert, die Informationen zum System vorhalten. Die Typdefinitionen der Variablen sind in der Bibliothek "SchleicherLib" im Abschnitt *PLC_Types* zu finden.

8.4 Bibliotheken und Funktionsbausteine in MULTIPROG

Funktionsbausteine sind in Bibliotheken zusammengefasst. Sie werden, je nach Steuerungstyp, beim Erstellen eines neuen MULTIPROG-Projektes automatisch eingebunden oder können bei Bedarf manuell eingebunden werden.

Wichtig!

Grundsätzlich muss die zum Steuerungsbetriebssystem passende, aktuellste Version der Bibliotheken eingesetzt werden.

Bibliotheken	XCA 11xx	XCN 7xx	XCS 7xx	XCN 5xx	XCS 5xx	XCN 3xx	XCS 3xx	MCS 2x	ProNumeric	ProSyCon	MCS 20-20	MCS 20-21	Simulation
PROCONOS	+	+	+	+	+	+	+	+	+	+	+	+	+
BIT_UTIL	+	+	+	+	+	+	+	+	+	+	+	+	+
CANopen_Vxxx	+	+	+	+	+	0	0	0	+	+	–	+	–
CFB_Vxxx	0	0	0	0	0	0	0	–	0	0	–	–	–
CNC_Vxxx	+	+	–	+	–	+	–	–	+	–	–	–	–
Date_Time	+	+	+	+	+	+	+	0	+	+	+	+	–
MC_Vxxx	0	+	–	+	–	+	–	–	–	–	–	–	–
Microline	-	–	–	–	–	–	–	+	–	–	+	+	–
MMI	+	0	0	0	0	0	0	-	0	0	0	0	–
PLC_Vxxx	+	+	+	+	+	+	+	0	+	+	–	–	–
PNS_Vxxx	-	–	–	–	+	+	–	–	–	–	–	–	–
Profibus_Vxxx	0	0	0	0	0	0	0	-	0	0	–	–	–
SchleicherLib_Vxxx	+	+	+	+	+	+	+	-	+	+	–	–	–
Serial	0	0	0	0	0	0	0	0	0	0	0	0	–
XCx7_Vxxx	+	+	+	–	–	–	–	–	–	–	–	–	–

\+ Werden beim Erstellen eines neuen Projektes automatisch eingebunden.
0 Können je nach Bedarf manuell eingebunden werden.
– Nicht möglich oder unnötig.

Tabelle 26: Bibliotheken und Funktionsbausteine in MULTIPROG

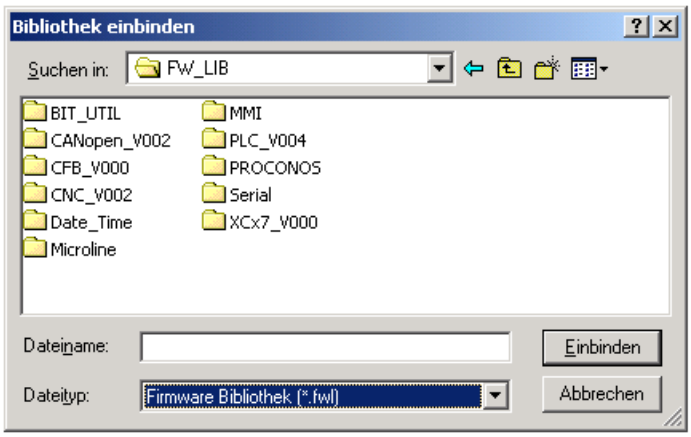
Funktionsbausteine können folgendermaßen eingebunden werden:

- Mit der rechten Maustaste im Projektbaum von MULTIPROG das Kontextmenü *Bibliotheken / Einfügen / Bibliothek* öffnen (Bild 87).

Bild 87: MULTIPROG, Kontextmenü "Bibliotheken"

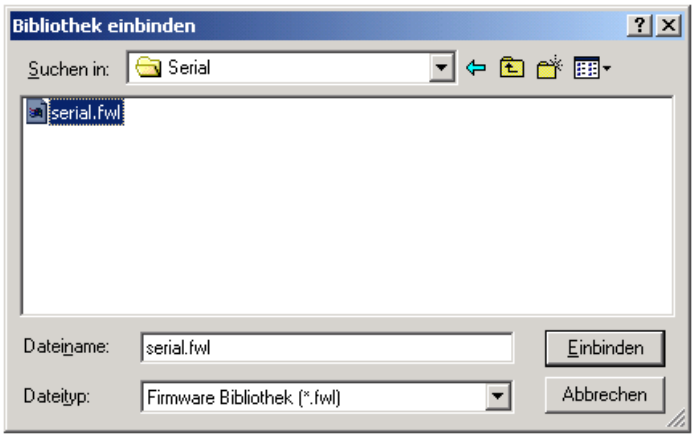
- Den Pfad *.\KWSoft\MWT\PLC\FW_LIB* und den Dateityp *Firmware Bibliothek (*.fwl)* wählen (Bild 88).

Bild 88:
MULTIPROG, Dialogfenster "Bibliothek einbinden"



Jede Bibliothek ist in einem eigenen Pfad gespeichert. Soll z.B. die Bibliothek *Serial* eingebunden werden, muss sie im gleichnamigen Pfad der Bibliothek ausgewählt werden (Bild 89).

Bild 89:
MULTIPROG, Einbinden der Bibliothek "Serial"



Zu den Bibliotheken (außer SchleicherLib) ist eine Online-Hilfe vorhanden. Die Online-Hilfe ist über das Kontextmenü der jeweiligen Bibliothek erreichbar. Das Kontextmenü wird aktiv, wenn mit der rechten Maustaste auf das Icon der Bibliothek geklickt wird.

8.4.1 Hinweis zu den Variablendeklarationen der Beispielprogramme von FBs

Die Beispielprogramme in den Hilfen zu den Funktionsbausteinen enthalten Variablendeklarationen nach IEC 61131-3 mit den Schlüsselwörtern VAR und END_VAR. Sollen die Beispielprogramme mit MULTIPROG angewendet werden, müssen die Variablen-deklarationen in Tabellenform auf dem Variablen-Arbeitsblatt der benutzten POE von Hand eingetragen werden.

8.4.2 Bibliothek CANopen_Vxxx

Die Bibliothek "CANopen_Vxxx" enthält Funktionsbausteine für die Parametrierung und Diagnose des CANopen-Netzwerkes.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
CO_NET_SDO_WRITE	150	sendet ein Service Data Object (SDO)	XCx ProNumeric ProSyCon MCS 20-21
CO_NET_SDO_READ	151	empfängt ein Service Data Object (SDO)	
CO_NET_GET_LOCAL_NODE_ID	152	liefert die eigene Node-ID zurück	
CO_NET_GET_STATE	153	liefert den aktuellen CANopen-Status	
CO_NET_GET_KERNEL_STATUS	154	liefert den sog. erweiterten CANopen-Kernelstatus	
CO_NET_NMT	155	setzt den Status eines oder aller Geräte im CANopen-Netzwerk	
CO_NET_RECV_EMY_DEV	156	liest etwaige Emergency-Nachrichten von einem bestimmten Netzwerk-Knoten	
CO_NET_RECV_EMY	157	liest etwaige Emergency-Nachrichten von einem beliebigen Netzwerk-Knoten	
CO_NET_RECV_ERR_DEV	160	liest etwaige Error-Nachrichten von einem bestimmten Netzwerk-Knoten	
CO_NET_RECV_ERR	161	liest etwaige Error-Nachrichten von einem beliebigen Netzwerk-Knoten	
CO_NET_SENDL2	162	sendet beliebige CAN Layer 2-Nachrichten	
CO_NET_PING	163	führt ein Ping auf einen bestimmten Netzwerk-Knoten aus	
CO_NET_RESTART_CAN	164	startet die CANopen Kommunikation neu (z.B. nach "bus-off")	
CO_NET_RESTART_ALL	165	startet den kompletten CANopen-Stack neu	
CO_NET_SHUTDOWN	166	stoppt den CANopen-Stack	
CO_NET_CAN_SYNC	170	ermöglicht die Synchronisation zwischen SPS-Task und den CANopen-Stack	

Tabelle 27: Bibliothek CANopen_Vxxx

8.4.3 Bibliothek CFB_Vxxx

Die an IEC 61131-5 angelehnte Bibliothek "CFB_Vxxx" enthält Funktionsbausteine zur Peer-to-Peer-Kommunikation über TCP/IP.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
CONNECT_V	60	stellt eine Peer-to-Peer-Verbindung zwischen zwei Teilnehmern her	XCx ProNumeric ProSyCon
USEND_V	61	sendet beliebige Daten	
URCV_V	62	empfängt beliebige Daten	

Tabelle 28: Bibliothek CFB_Vxxx

8.4.4 Bibliothek CNC_Vxxx

Die Bibliothek "CNC_Vxxx" enthält Funktionsbausteine für das Lesen und Schreiben von Systemdaten, sercos-, XRIO- und CAN-Antriebsparametern und PROFIBU-DP-Antriebsparametern.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
READ_Q_PARAM_*	200 bis 207	liest einen CNC-Systemdaten-Parameter	XCN ProNumeric
WRITE_Q_PARAM_*	208 bis 215	schreibt einen CNC-Systemdaten-Parameter	
SAVE_Q_PARAM_*	221	speichert die CNC-Systemdaten-Parameter auf der Festplatte	
SAVE_R_PARAM_*	220	speichert die CNC-Rechenparameter auf der Festplatte	
READ_SERC_PARAM	302	liest einen sercos-Parameter	
WRITE_SERC_PARAM	303	schreibt einen sercos-Parameter	
SET_SERC_PHASE	304	Umschaltung der sercos-Kommunikationsphase	
SET_SERC_COMMAND	308	ausführen eines sercos-Kommandos	
MC_ANALOG	300	XRIO Motion Control Baustein (mit Lageregler)	XCN
MC_ANALOG_1_AXIS	307	XRIO Motion Control Baustein für eine Achse (mit Lageregler)	
READ_AXIS_PAGE	305	liest einen Parameter aus der einer Achse zugeordneten sogenannten Remotepage.	
WRITE_AXIS_PAGE	306	schreibt einen Parameter in die einer Achse zugeordneten sogenannten Remotepage	
READ_MF	230	Dieser CNC Funktionsbaustein liest Einträge der M-Funktionsqueue	
MC_CAN	301	CAN MotionControl Baustein	Nicht XCN700 XCN1100
MC_DP	309	PROFIBUS-DP Motion Control Baustein	
MC_DP_1_AXIS	310	PROFIBUS-DP Motion Control Baustein für eine Achse	

Tabelle 29: Bibliothek CNC_Vxxx

Die Funktionsbausteine READ_AXIS_PAGE und WRITE_AXIS_PAGE sind von der Bibliothek XCx7_Vxxx in die Bibliothek CNC_Vxxx übernommen worden (ab CNC_V006 / XCx7_V002).

8.4.5 Bibliothek Date_Time

Die XCx verfügt über eine gepufferte Echtzeituhr mit Kalender (Berücksichtigung von Schaltjahren) und einer Auflösung von 1 s.

Datum und Uhrzeit können mit den Funktionsbausteinen aus der Bibliothek "Date_Time" gelesen und gesetzt werden.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
GET_TIME	130	Zeit lesen	XCx
GET_DATE	128	Datum lesen	ProNumeric
SET_TIME	131	Zeit setzen	ProSyCon
SET_DATE	129	Datum setzen	MCS xx-xx

Tabelle 30: Bibliothek Date_Time

8.4.6 Bibliothek MC_Vxxx

Die Bibliothek "MC_Vxxx" (Motion Control) enthält Funktionsbausteine zur Programmierung von Bewegungsabläufen in der SPS.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
MC_MoveAbsolute	320	Achse wird beauftragt, auf eine absolute Position zu fahren	CXN 300 XCN 5xx XCN 700
MC_MoveRelative	321	Achse wird beauftragt, eine Strecke zu fahren	
MC_MoveAdditive	322	Achse wird beauftragt, auf eine absolute Position zu fahren	
MC_MoveVelocity	324	Achse wird beauftragt, mit der vorgegebene Geschwindigkeit zu fahren	
MC_Home	325	Achse wird beauftragt, zu referenzieren	
MC_Stop	326	Achse wird beauftragt, Achsbewegung zu beenden	
MC_Power	327	Achse wird beauftragt, Drehmoment (Reglerfreigabe) einzuschalten	
MC_ReadStatus	328	Status-Informationen der Achsen werden gelesen	
MC_ReadAxisError	329	Aktuelle Fehlernummer wird gelesen	
MC_Reset	330	Reset (Fehlerquittierung) wird durchgeführt	
MC_ReadParameter	331	Ein Parameter der Achse wird gelesen	
MC_ReadBoolParameter	332	Ein boolscher Parameter der Achse wird gelesen	
MC_WriteParameter	333	Ein Parameter der Achse wird geschrieben	
MC_WriteBoolParameter	334	Ein boolscher Parameter der Achse wird geschrieben	
MC_ReadActualPosition	335	Aktuelle Achsposition wird gelesen	
MC_GetCncAxis	345	Achse wird von der CNC ausgeliehen, um sie in der SPS verfahren zu können	
MC_ReleaseCncAxis	346	Ausgeliehene Achse wird zur CNC zurückgegeben	

Tabelle 31: Bibliothek MC_Vxxx

8.4.7 Bibliothek MMI

Die Bibliothek "MMI" realisiert die Kommunikation mit einem Bediengerät der COP-Familie über die serielle Schnittstelle der Steuerung.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
PPF_COP_COMM	140	kommuniziert mit einem COP-Bediengerät (PNet-Protokoll)	XCx ProNumeric ProSyCon MCS xx-xx

Tabelle 32: Bibliothek MMI

8.4.8 Bibliothek PLC_Vxxx

Über den Umfang der Standard IEC- bzw. ProConOS-Funktionsbausteine hinaus werden weitere, steuerungsspezifische Firmware-Funktionsbausteine in dieser Bibliothek bereitgestellt.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
PUT_ERROR	400	erzeugt eine nutzerdefinierte Fehlermeldung (bitte nicht mehr verwenden)	XCx ProNumeric ProSyCon
PUT_ERROR2	401	erzeugt eine nutzerdefinierte Fehlermeldung	
CLEAR_ERROR	402	löscht eine mit Lock-Flag abgesetzte Fehlermeldung	
READ_FILE	405	lesender Dateizugriff	
WRITE_FILE	406	schreibender Dateizugriff	
SEND_MAIL	410	sendet eine E-MAIL (SMTP Client)	
XFIO_CONFIG	420	XFIO Interrupt Konfiguration	XCx
XFIO_TIMER_INT_CONFIG	421	XFIO timergesteuerte Interrupt Konfiguration	
XRIO_STATE	422	XRIO Statusinformationen	
GET_MTS	430	liefert den aktuellen Zeitwert in µs-Ticks	XCx ProNumeric ProSyCon
OPEN_PROFILE	431	öffnet eine Datei im INI-Format	
NEW_PROFILE	432	legt eine neue Datei im INI-Format an	
FLUSH_PROFILE	433	schreibt aktualisierte Datei im INI-Format	
CLOSE_PROFILE	434	schließt eine Datei im INI-Format	
GET_PROFILE_STRING	435	liest einen String aus einer Datei im INI-Format	
GET_PROFILE_INT	436	liest einen Integer-Wert aus einer Datei im INI-Format	
GET_PROFILE_REAL	437	liest einen Real-Wert aus einer Datei im INI-Format	
WRITE_PROFILE_STRING	438	schreibt einen String in eine Datei im INI-Format	
WRITE_PROFILE_INT	439	schreibt einen Integer-Wert in eine Datei im INI-Format	
WRITE_PROFILE_REAL	440	schreibt einen Real-Wert in eine Datei im INI-Format	

Tabelle 33: Bibliothek PLC_Vxxx

8.4.9 Bibliothek PNS_Vxxx

Die Bibliothek "PNS_Vxxx" enthält Funktionsbausteine für die Parametrierung und Diagnose des PROFINET-Netzwerkes.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
PNSReadIOData	530	liest IO-Daten	XCx 5xx
PNSWriteIOData	531	schreibt IO-Daten	
PNSCommunicating	532	liefert Status von PROFINET-Verbindung	

Tabelle 34: Bibliothek PNS_Vxxx

8.4.10 Bibliothek Profibus_Vxxx

Die Bibliothek "Profibus_Vxxx" enthält Funktionsbausteine für die Kommunikation über die PROFIBUS-Karte.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
DP_NET_GET_STATE	190	liefert den Status der PROFIBUS-Karte	XCx
DP_NET_PUT_MSG	191	setzt eine Nachricht an das Message-Interface der Hilscher-Karte ab	ProNumeric
DP_NET_GET_MSG	192	holt eine Nachricht vom Message-Interface der Hilscher-Karte ab	ProSyCon

Tabelle 35: Bibliothek Profibus_Vxxx

8.4.11 Bibliothek SchleicherLib_Vxxx

Die Bibliothek "SchleicherLib_Vxxx" enthält Datentypdefinitionen der Firmware, die für MULTIPROG bereitgestellt werden. Funktionsbausteine sind in dieser Bibliothek nicht enthalten.

8.4.12 Bibliothek Serial

Die Bibliothek "Serial" enthält Funktionsbausteine für die serielle Kommunikation der Steuerungen.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
PORT_OPEN	135	öffnet eine serielle Schnittstelle	XCx
PORT_CLOSE	136	schließt eine serielle Schnittstelle	ProNumeric
PORT_READ	137	gibt Zeichen auf einer seriellen Schnittstelle aus	ProSyCon
PORT_WRITE	138	liest Zeichen von einer seriellen Schnittstelle	MCS xx-xx
PORT_STATE	139	liefert Statusinformation einer seriellen Schnittstelle	

Tabelle 36: Bibliothek Serial

8.4.13 Bibliothek XCx7_Vxxx

Die Bibliothek "XCx7_Vxxx" enthält Funktionsbausteine die ausschließlich bei den Steuerungstypen XCx 700 und XCA verwendet werden.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
UZB_VR	250	Funktionsbaustein für den Betrieb der UZB 2VR Module	XCx 7xx XCA
UBA_ERR_CTRL	251	Fehlerhandling der UBA-Erweiterungsmodule	
READ_RP	252	lesender Zugriff auf System-U Remotepages	
WRITE_RP	253	schreibender Zugriff auf System-U Remotepages	
IBSM	254	InterBus-S Master (USK-DIM)	

Tabelle 37: Bibliothek XCx7_Vxxx

8.5 Das SPS-Betriebssystem ProConOS

8.5.1 Die Initialisierungsdatei ProConOS.INI

Mit Hilfe der Datei ProConOS.INI lassen sich erweiterte Einstellungen (z.B. der Kommunikationstreiber, der Systemtasks und des CANopen-Stacks) applikationsspezifisch anpassen. Falls ProConOS.INI noch nicht existiert oder gelöscht wurde, wird die Datei initial mit Default-Werten beim Start der Steuerungssoftware angelegt.

Pfad der Datei auf dem Compact Flash: */ata0/OS/PLC/ProConOS.INI*

8.5.2 Beschreibung der ProConOS.INI Section- und Key-Einträge

Section PLC

```
[PLC]
; max. Größe des PLC Programmspeichers: 512…12288 kByte
PC_PROGRAM_SIZE =4096

; verwende den ProConOS Socket Kommunikations- Treiber
PC_SOCKET_DRV  = 1      ; yes = 1 (default), no = 0
; max. Anzahl der ProConOS- Clients
; bei gleichzeitigen Zugriff auf die Steuerung
PC_SOCKET_BLOG = 4      ; default

; Verwende seriellen ProConOS- oder HBG-Kommunikations-Treiber
; no = 0, ProConOS protocol = 1, HBG protocol = 2, DriveTop = 3
PC_SERIAL0_DRV = 0      ; RS 232
PC_SERIAL0_BR  = 19200  ; Baudrate = 19200 (default)
PC_SERIAL1_DRV = 0      ; RS 422 / RS 485
PC_SERIAL1_BR  = 19200  ; Baudrate = 19200 (default)
PC_SERIAL2_DRV = 0      ; RS 232, XCx7 only
PC_SERIAL2_BR  = 19200  ; Baudrate = 19200 (default, RS 232, XCx7 only)

PC_SERIAL_DELAY = 100   ; ProConOS driver receive delay (2 .. 200ms)
```

Section CNC

```
z.Z. ohne Eintrag (obsolet)
```

Section CAN

```
[CAN]
; 1 =enable (default), 0 = disable CAN driver
CAN_ENABLE  =0

; Priorität der CANopen Process Task
CAN_PRIO_HIGH = 35      ; 35 = default (1..200)
; Priorität der CANopen Interrupt Handling Task
CAN_PRIO_INT  = 10      ; 10 = default (0..20)

; Restart den CANopen Prozess nach PLC STOP (NMT master!)
RESTART_CAN    = 0      ; yes=1, no=0 (default)

; PLC STOP nach CAN heartbeat error
HBE_STOP_PLC   = 1      ; yes = 1 (default), no = 0

; PLC STOP nach CAN Bus Off
CBO_STOP_PLC   = 0      ; yes = 1 (default), no = 0

; obsolet, ohne Funktion!! - starte die CANopen Task mit hoher Priorität
CAN_HIGH_PRIO = 0       ; yes = 1, no=0 (default)
```

Section CANxx

```
; CAN parameter Definitionen:
; Achtung: Überschreibt etwaige Einstellungen in der CANconf.dat!!
[CAN00] ; e.g. card# 0
; any description
DESCRIPTION     = CANconf #0 (XCx 500 series)
NODE_ID         = 1     ; bus address, 1..127
; baudrate
; (1=10,2=20,3=50 obsolet),4=125,5=250,6=500,7=800,8=1000 MBit/s
BAUDRATE        = 4
BOOTUP_DELAY    = 0     ; bootup delay, 0..60 s
CYCLE_TIME      = 4     ; CAN cycle time, 0..255 ms
```

Section sercos

```
[sercos]
; default IP Address for NRT communication
sercos_IP=192.168.0.1
```

Section DPxx

```
; DP parameter Definitionen:
[DP00]                  ; e.g. card# 0
USE_COM_DRV         = 1                 ; use the simple COM slave driver,
                        ; 1 = yes (default), 0 = no
; any description
DESCRIPTION         = Profibus DPS #0 (XCx5 series)
BUS_ADDR            = 2                 ; 0..126
MASTER_FCONF        = 1                 ; the master force slaves
configuration,
                        ; no further configuration necessary
                        ; 1 = yes (default)/ 0 = no
FirmwareFileFolder =”/ata0/OS/DP/”      ; location of the firmware file
terminated
                ; with ”/” (for automated updates)
```

Section IODriver (CIF Driver für DPM Karten)

```
(s. ProConOS Manual for Hilscher CIF 30/50 and ProConOS CIF Driver Manual
for Hilscher fieldbus CIF interfaces)

[IODriver]      ; Name of section
; The 0 at the end of the parameter name characterises the first of n
possible
; further drivers IODriver[n]. The name refers to the current version of the
; driver. Thus in future versions only this place has to be changed but not
the
; user projects (see I/O configuration).
IODriver0       = "CIF_KW_V2.0"

[CIF_KW_V2.0@0]         ; One instance for one CIF board.
; If this key is set, the hardware communication  must be started and
stopped
; manually by the user. Specific function blocks are provided for this
purpose.
; Otherwise the driver will start the hardware communication at PLC_RUN and
; stop the hardware communication at PLC_STOP automatically.
StartUpManual   = 0

; If this key is set, the Hilscher configuration tool SyCon is able to
connect to
; and to configure installed CIF boards via TCP/IP. The TCP/IP address is
the
; same as of ProConOS. This key can only be set in the section of the first
CIF
; board.
ComServer       = 0
```

```
; This key determines the bus type of the installed boards. All installed
boards
; must be of the same bus type. This key can only be set in the section of
the first board.
BusTyp          = ”USR”         ; The proper Bus type for XCx controller.

; The default mode is "HostControlledBuffered". If there is no ComMode entry
; in the PROCONOS.INI file the default settings are applied. If ComMode =
; "NoChangeOfMode" the actual CIF card mode is applied. In this case the
; mode can be changed e.g. by application of the SyCon tool.
; "DirectDeviceControlled"      Direct Data Transfer, DEVICE Controlled
; "BufferedDeviceControlled"    Buffered Data Transfer, DEVICE Controlled
; "UncontrolledDirect"  Uncontrolled Direct Data Transfer
; "HostControlledBuffered"      HOST Controlled, Buffered Data Transfer
; "HostControlledDirect"        HOST Controlled, Direct Data Transfer
; "NoChangeOfMode"      Keep the mode set by Sycon,
;       needs to be set for every board
ComMode         = "HostControlledDirect"

; The Startup DPM configuration.
ConfigurationFile="/ata0/OS/DP/DPMconf0.dbm"
; The COM module firmware path - location of the firmware file terminated
with ”/”!
FirmwareFileFolder="/ata0/OS/DP/"
More then one board using:
If more than one board are placed in the respective hardware, or more than
one board are assigned in the
ProConOS-IO-Groups, an assignment for the hardware board and ProConOS-IO-
Group-Board is necessary.

[CIF_KW_V2.0@0]         ; possible are up to four instances of the same
driver: @0 ... @3.

;1 = CIF hardware driver assigns board number, only possible in section of
card 0
ManualBoardAssign = 1

; The device number of the card can be found on the card used. Also it is
possible to read out this with the SyCon.
DeviceNr    = 10504000
; The series number of the card can be found on the card used. Also it is
possible to read out this with the SyCon.
SerNr           = 00003930

[CIF_KW_V2.0@1]         ; possible are up to four instances of the same
driver: @0 ... @3.
; The device number of the card can be found on the card used. Also it is
possible to read out this with the SyCon.
DeviceNr    = 10304100
; The series number of the card can be found on the card used. Also it is
possible to read out this with the SyCon.
SerNr           = 00005648
```

Wichtig!

Das Lesen der Einträge aus den Sections [IODriver] und [CIF_KW_V2.0@0] verwendet andere Betriebssystem-Routinen. Vor dem "=" dürfen keine Leerzeichen stehen, z.B.:
nicht ManualBoardAssign = 1
sondern ManualBoardAssign=1

9 Das Multi-Task-System

9.1 Übersicht

Basis ist ein Echtzeit-Betriebssystem, das durch Taskprioritäten gesteuert wird. Im Programmiersystem MULTIPROG wird ein Programm einer Task zugewiesen. Den Tasks wiederum werden unterschiedliche Prioritätsstufen und Zeiten zugeordnet, die die Reihenfolge und Dauer der Bearbeitung gemäß ihrer Wichtigkeit sicherstellen.

Drei Prioritätsstufen für Tasks werden verwendet (Bild 90):

Bild 90: Multi-Task-System, Prioritätsstufen

- **Überwachungstask**
  Die Überwachungstask ist eine besonders geschützte Betriebssystemtask, die in der höchsten Prioritätsstufe abgearbeitet wird. Sie ermittelt Fehler, wie z.B. eine Division durch Null oder die Überschreitung der Ausführungszeit einer Task, und aktiviert die entsprechende Betriebssystemtask.
- **Anwender- und Defaulttasks**
  Auf der Anwender- und Defaulttask-Stufe laufen alle Tasks, die vom Anwender eingefügt werden. In diesem Bereich laufen auch einige wichtige Firmware-Tasks, die beim Parametrieren der Anwender-Tasks berücksichtigt werden müssen. Siehe dazu Abschnitt "Task-Prioritäten", Seite 113. Die Anwendertasks sind zeitüberwacht (Watchdog).
  - **Zyklische Tasks**
    führen die ihnen zugewiesenen Programme innerhalb eines definierten Zeitintervalls mit einer vom Anwender vorgegebenen Priorität aus. Die Task mit der höchsten Priorität wird als erste aufgerufen.
  - **Ereignistasks**
    werden vom Betriebssystem der Steuerung gestartet, wenn bestimmte Ereignisse wie z.B. Interruptsignal, CANopen- oder IPO-Task auftreten.
  - **Defaulttask**
    ist die Anwendertask mit der niedrigsten Priorität. Sie ist nicht zeitüberwacht und wird als Hintergrundtask aktiviert, wenn zum entsprechenden Zeitpunkt keine höherprioren Anwendertasks aktiv sind.
- **Betriebssystemtasks**
  Auf der Prioritätsstufe für Betriebssystemtasks laufen Tasks z.B. für Kommunikation, Debugging, Speicherverwaltung und Systemkontrolle vom Anwender unbeeinflusst ab.

9.2 Anwender-Tasks

Achtung!

Eine falsche oder zumindest ungeeignete Wahl der Anwender-Task-Einstellungen hinsichtlich Typ, Priorität oder Interrupt-Mode usw. - insbesondere in Kombination mit langen Programmlaufzeiten - kann zu Steuerungsfehlfunktionen führen, da essentielle Betriebssystem-Tasks verdrängt werden.

Beschreibung im Abschnitt Task-Prioritäten unbedingt beachten.

Anwender-Tasks sind alle Tasks, die durch den Anwendungsprogrammierer eingefügt werden.

Die Default-Task gehört ebenfalls zur Prioritätsstufe für Anwender-Tasks. Sie ist die Anwender-Task mit der niedrigsten Priorität. Die Default-Task wird abgearbeitet, wenn zum entsprechenden Zeitpunkt keine Anwender-Task aktiv ist.

Es können verschiedene Anwender-Tasktypen verwendet werden.

9.2.1 Zyklische Tasks

Zyklische Tasks führen die ihnen zugewiesenen Programme innerhalb eines definierten Zeitintervalls mit einer vom Anwender vorgegebenen Priorität aus.

In MULTIPROG können den einzelnen Tasks Prioritäten zwischen 0 und 31 zugeordnet werden. 0 steht für die höchste, 31 für die niedrigste Priorität. Die Task mit der höchsten Priorität wird als erste aufgerufen. Die User-Task-Prioritäten werden auf die Prioritätsstufen des Echtzeitbetriebssystems abgebildet (siehe Abschnitt Task-Prioritäten).

Wenn die Watchdog-Zeit einer zyklischen Task höher ist als die eingestellte Intervallzeit und die Ausführung der Task nicht beendet ist, bevor die eingestellte Intervallzeit erreicht wird, werden ein oder mehrere Ausführungs-Zyklen übersprungen.

9.2.2 Ereignis-Tasks

Ereignis-Tasks oder auch Event-Tasks werden vom Betriebssystem gestartet, wenn bestimmte Ereignisse auftreten. Gegenwärtig sind folgende Ereignisse definiert:

Interne Bezeichnung	Ereignis-Nr.	Bemerkungen
Interrupts		
PLC_EVENT_XFIO_I0	0x00	XFIO Interrupt (Input 0, XCx 3/5)
PLC_EVENT_XUIO_0	0x00	U-Bus Interrupt 0 (XCx7, UBE32 0,1I Eingang 0)
PLC_EVENT_XFIO_I1	0x01	XFIO Interrupt (Input 1, XCx 3/5)
PLC_EVENT_XUIO_1	0x01	U-Bus Interrupt 1 (XCx7/11, UBE32 0,1I Eingang 1)
PLC_EVENT_XUIO_2	0x02	U-Bus Interrupt 2 (XCx7/11, UBE32 0,1I Eingang 2)
PLC_EVENT_XUIO_3	0x03	U-Bus Interrupt 3 (XCx7/11, UBE32 0,1I Eingang 3)
Synchronisation		
PLC_EVENT_POS	0x04	Lageregler-Task (nur XCN)
PLC_EVENT_CAN	0x05	- CANopen-Task, - gilt auch für Profibus-Task (microLine, XCx micro)
PLC_EVENT_IPO	0x06	CNC IPO-Task (nur XCN)
PLC_EVENT_DECO	0x07	CNC DECO-Task (nur XCN)
PLC_EVENT_MCSIO	0x08	MCS / XCS20 IO-Treiber Synchronisation (microLine, XCx micro)
Reserviert	0x09	
PLC_EVENT_XFIO_I10	0x0A	Messinterrupt aktiv 0 (nur XCx3/5)
PLC_EVENT_XFIO_I11	0x0B	Messinterrupt aktiv 1 (nur XCx3/5)
Reserviert	0x0C	
Reserviert	0x0D	
Reserviert	0x0E	
PLC_EVENT_AC_FAIL	0x0F	AC Fail (XCx 11, ProNumeric)

Tabelle 38: Multi-Task-System, Ereignis-Tasks

Die Ereignisnummer wird in der Taskeinstellung von MULTIPROG verwendet, um das Ereignis zu spezifizieren, das die Ereignis-Task startet.

Die vorgegebene Priorität wird, außer bei gesetzter Bypass-Option, vom System berücksichtigt. (Bypass hebt den normalen Taskwechsel auf, sodass die zugewiesenen Programme sofort ausgeführt werden, wenn das Ereignis eintritt.)

Es werden bis zu 16 Ereignisse in eine Warteschlange gesetzt. Diese Ereignisse gehen daher nicht verloren und werden später ausgeführt. Dies gilt auch im Falle eines Auftretens neuer Ereignisse vor der Ausführung der zugewiesenen Ereignis-Task.

9.2.3 System-Tasks

System-Tasks bzw. Systemprogramme (SPGs) werden automatisch vom Betriebssystem gestartet, wenn im Zusammenhang mit dem Betriebssystem ein Ereignis auftritt. Verschiedene SPGs sind verfügbar, wie in der folgenden Tabelle dargestellt:

Nr.	Name	Ereignis	Aktionen
SPG 0	**WARM_START**	wird bei einem Warmstart ausgeführt	• remanente Daten werden nicht initialisiert • nicht-gepufferte Daten werden initialisiert • die Open-Funktion des I/O-Treibers wird ausgeführt • Anwender-Tasks werden aktiviert • SPS wechselt in den Zustand 'Betrieb'
SPG 1	**COLD_START**	wird bei einem Kaltstart ausgeführt	• alle Daten werden initialisiert • die Open-Funktion des I/O-Treibers wird ausgeführt • Anwender-Tasks werden aktiviert • SPS wechselt in den Zustand 'Betrieb'
SPG 2	**TO_STOP**	wird ausgeführt, wenn die Programmausführung gestoppt wird	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 10	**WATCHDOG**	wird ausgeführt, wenn die Ausführung einer Task nicht innerhalb ihrer Watchdogzeit beendet ist	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 11	**ZERODIV**	wird ausgeführt, wenn während der Programmausführung eine Division durch Null aufgetreten ist	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 12	**STACKOVER**	wird ausgeführt, wenn ein Stacküberlauf aufgetreten ist. Wird nur ausgeführt, wenn das Kontrollkästchen 'Stack-Prüfung' im Dialog 'Ressource ... einrichten' in MULTIPROG aktiviert wurde.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 13	**BADCAL**	wird ausgeführt, wenn eine herstellerspezifische POE aufgerufen wird, die nicht existiert	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 14	**IOERROR**	wird ausgeführt, wenn ein Fehler im I/O-Treiber auftritt, während der Prozess abläuft	• SPS setzt Abarbeitung fort
SPG 16	**MATHERR**	wird ausgeführt, wenn ein Gleitkommafehler in einer arithmetischen Funktion auftritt	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'

Nr.	Name	Ereignis	Aktionen
SPG 17	CPU_OVERLOAD	wird ausgeführt, wenn eine CPU-Überlastung auftritt	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 18	INITIODRV_ERR	wird ausgeführt, wenn beim Initialisieren des I/O-Treibers während eines Kalt- oder Warmstarts ein Fehler auftritt	• SPS wird nicht gestartet
SPG 19	BOUNDS_ERR	wird ausgeführt, wenn die Grenzen eines Felds oder einer Struktur überschritten wurden. Wird nur ausgeführt, wenn das Kontrollkästchen 'Index-Prüfung' oder das Kontrollkästchen 'Feldbegrenzungs-Prüfung' im Dialog 'Ressource ... einrichten' in MULTIPROG aktiviert wurde.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 20	BUS_ERR	wird ausgeführt, wenn Variablen mit einem Datentyp ≥ 2 Bytes und ungeraden Adressen verwendet wurden oder wenn in MULTIPROG ein interner Fehler aufgetreten ist. Nur bei Motorola-Plattformen.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 21	STRING_ERR	wird ausgeführt, wenn ein Fehler bei einer Zeichenfolge-Operation auftritt, z.B. wenn eine Zeichenfolge durch eine andere ersetzt werden sollte, aber nicht gefunden wurde.	• Das Verhalten einer Zeichenfolge-Ausnahme hat sich geändert! In der Standardeinstellung wird nach dem Auftreten einer Zeichenfolge-Ausnahme das SPG 21 aufgerufen. Zusätzlich wird ein Eintrag in den Fehlerkatalog vorgenommen, der die Modul- und Zeilennummer enthält. Die SPS bleibt im 'RUN'-Status.

Tabelle 39: Multi-Task-System, System-Tasks

Hinweis

System-Tasks werden nicht vom Watchdog überwacht.

9.2.4 Default-Task

Die Default-Task läuft mit der niedrigstmöglichen Anwender-Task-Priorität als sog. Hintergrund-Task und ist nicht zeitüberwacht. Sie wird dann aktiviert, wenn alle höherprioren Anwender-Tasks abgearbeitet wurden. Die Default-Task ist dabei so konfiguriert, dass sie nur einen Teil der zur Verfügung stehenden Restzeit beansprucht. In jeder Ressource ist nur eine Default-Task erlaubt. Es wird empfohlen, ausschließlich zyklische Tasks zu verwenden.

Hinweis

Alle Treiber der I/O-Konfiguration, die nicht explizit einer Anwender-Task zugeordnet wurden, führen zum automatischen Anlegen der Default-Task und werden im Kontext der Default-Task ausgeführt.

9.3 Anwender-Task-Information

Für jede Anwender-Task werden Informationen auf System-Variablen abgebildet. Die unten abgebildeten Typdefinitionen der Systemvariablen sind in der Bibliothek "SchleicherLib" im Abschnitt *PLC_Types* zu finden.

Typdefinition	Bemerkung
`TYPE`	
`  TaskNameType  : ARRAY [1..10] OF BYTE;`	
`END_TYPE`	
`TYPE`	
`  TaskInfoType0 : STRUCT`	
`    MaxTask      : INT;          (* 00: *)`	max. mögliche Taskanzahl
`    CurTask      : INT;          (* 02: *)`	aktuelle Taskanzahl
`  END_STRUCT (* TaskInfoType0 *);`	
`END_TYPE`	
`TYPE`	
`  TaskInfoType1 : STRUCT`	
`    TaskName     : TaskNameType; (* 04: *)`	Taskname
`    TaskPrio     : INT;          (* 14: *)`	Taskpriorität
`    TaskMode     : INT;          (* 16: *)`	Taskmode
`    TaskPeriod   : INT;          (* 18: [ms] *)`	Taskperiode in ms
`    TaskStack    : INT;          (* 20: *)`	Größe des benutzten Task-Stacks
`    MainPoe      : INT;          (* 22: assigned PLC program *)`	zugeordnetes SPS-Programm
`    TaskWatchDog : INT;          (* 24: [ms] *)`	Watch-Dog-Zeit in ms
`    reserve0     : DINT;         (* 26: *)`	
`    MaxStack     : INT;          (* 30: max. used stack *)`	Größe des möglichen Task-Stacks
`    CurDuration  : INT;          (* 32: [ticks] *)`	aktuelle Taskdauer einschließlich bevorrechtigter Aufrufe
`    MinDuration  : INT;          (* 34: [ticks] *)`	minimale Taskdauer
`    MaxDuration  : INT;          (* 36: [ticks] *)`	maximale Taskdauer
`    AveDuration  : INT;          (* 38: [ticks] *)`	mittlere Taskdauer
`    CurDelay     : INT;          (* 40: [ticks] *)`	aktuelle Taskverzögerung
`    MinDelay     : INT;          (* 42: [ticks] *)`	minimale Taskverzögerung
`    MaxDelay     : INT;          (* 44: [ticks] *)`	maximale Taskverzögerung
`    AveDelay     : INT;          (* 46: [ticks] *)`	mittlere Taskverzögerung
`  END_STRUCT (* TaskInfoType1 *);`	
`END_TYPE`	

Die Variablen werden mit den Typen *TaskInfoType0* und *TaskInfoType1* deklariert (Bild 91).

Bild 91:
Multi-Task-System, Variablendeklaration

Name	Typ	Verwendung	Beschreibung	Adresse	Anfangsw...	Reman...
TaskInfo0	TaskInfoType0	VAR_GLOBAL		%MD 1.1000		
TaskInfo1	TaskInfoType1	VAR_GLOBAL		%MD 1.1004		
TaskInfo2	TaskInfoType1	VAR_GLOBAL		%MD 1.1068		
TaskInfo3	TaskInfoType1	VAR_GLOBAL		%MD 1.1132		
TaskInfo4	TaskInfoType1	VAR_GLOBAL		%MD 1.1196		
TaskInfo5	TaskInfoType1	VAR_GLOBAL		%MD 1.1260		

Die folgende Anwendertask-Information wird mit einem Offset von 64 ab 1004 deklariert (1004 + 64 = 1068 usw.).

Die Reihenfolge der Tasks wird durch den Rang der Task im Projektbaum *Physikalische Hardware/Configuration/Resource/Tasks* festgelegt.

9.4 Task-Prioritäten

Die Tabelle gibt eine Übersicht über die empfohlenen Task-Prioritäten bzw. deren Einordnung hinsichtlich wichtiger reservierter Firmware-Tasks (*tfwLAGE, tfwCANhigh, tfwIPO*).

Warnung!

Eine falsche oder zumindest ungeeignete Wahl der Anwender-Task-Einstellungen hinsichtlich Typ, Priorität oder Interrupt-Mode usw. – insbesondere in Kombination mit langen Programmlaufzeiten – kann zu Steuerungsfehlfunktionen führen, da essentielle Firmware-Tasks (*tfwLAGE, tfwCANhigh, tfwIPO*) verdrängt werden.

Die Taskzuordnung und Taskzeiteinstellung überprüfen bzw. anpassen.

MULTIPROG-Priorität	RTOS*-Priorität (default)	RTOS* Task-Name	Verwendung
0	30	beliebig	z.B. Anwender-Task (Ereignis 0)
1	31	beliebig	z.B. Anwender-Task (Ereignis 1)
2	32	beliebig	z.B. Anwender-Task (Ereignis 4)
3	33	*tfwLAGE*	reserviert für Lageregel-Task (nur XCN)
4	34	beliebig	z.B. Anwender-Task (Ereignis 4, 5)
5	35	*tfwCANhigh*	reserviert für CAN- Stack Task (Option CAN_HIGH_PRIO = 1)
6	36	beliebig	z.B. Anwender-Task (Ereignis 5, 6)
7	37	*tfwIPO*	reserviert für IPO-Task (nur XCN)
8	38	beliebig	z.B. Anwender-Task (Ereignis 5)
9	39	*tfwCANhigh*	reserviert für CAN-Stack-Task (Option CAN_HIGH_PRIO = 0)
10	40	beliebig	z.B. Anwender-Task (Ereignis 5)
11..15	41..45	beliebig	z.B. zyklische Anwender-Tasks
16..31	46	beliebig	z.B. sonstige, zyklische Anwender-Tasks
Default	127	*default*	Hintergrund-Task
			*Real Time Operating System

Tabelle 40: Multi-Task-System, Taskprioritäten

Hinweis

Das System unterstützt insgesamt 18 Anwender-Tasks (Prioritätsstufen 0..16 und die Default-Task). Tasks mit Prioritätswerten ≥ 16 werden mit Priorität 16 ausgeführt.

9.5 Tasks und Watchdogs

Es gibt zu jeder anwenderdefinierten Task einen eigenen einstellbaren Watchdog. Der Watchdog überprüft, ob die Taskausführung am Ende des Watchdog-Zeitintervalls beendet ist. Wenn die Taskausführung nach dieser Zeit nicht beendet wird, wird die System-Task *SPG 10 'WATCHDOG'* ausgeführt und die SPS geht in den *'STOP'*-Zustand über, wenn keine weiteren Aktionen programmiert wurden. Zusätzlich wird ein Eintrag in den Fehlerkatalog vorgenommen. Das Watchdog-Zeitintervall beginnt, wenn die Task bereit für die Ausführung ist. Das Watchdog-Zeitintervall wird im Dialog *'Task ... einrichten'* in MULTIPROG festgelegt.

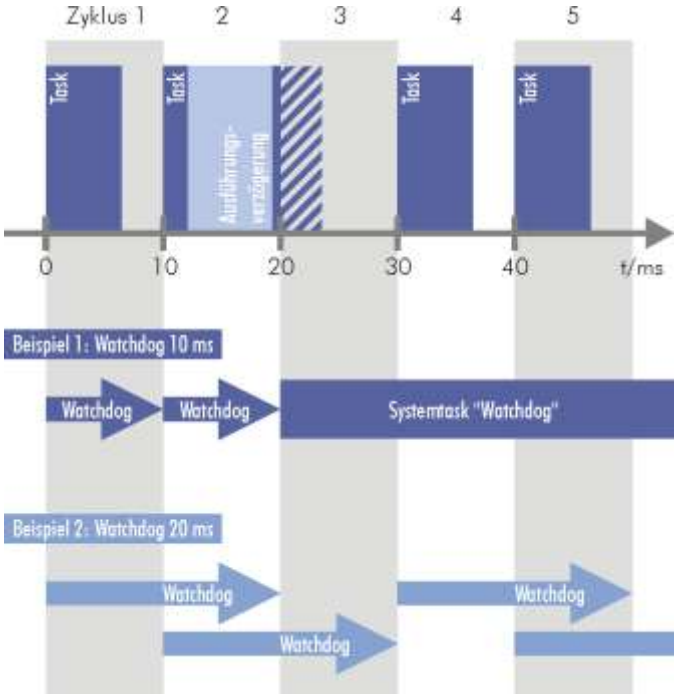
Hinweis

Wenn die Ausführungsdauer der Task sowie die Watchdog-Zeit annähernd denselben Wert haben und eine hohe CPU-Auslastung vorliegt, ist es möglich, dass während der Umsetzung einiger Online-Bedienschritte die Watchdog-Zeit überschritten wird.

Ein Grund für dieses Verhalten kann sein, dass während des Debuggens im Online-Modus der Adressstatus mit Durchlaufkontrolle ausgewählt wurde.

Beispiel

Bild 92: Multi-Task-System, Beispiel für Tasks und Watchdogs



Im **Beispiel 1** ist die Watchdog-Zeit der angezeigten Task auf 10 ms eingestellt. In der Abbildung überschreitet sie ihre Watchdog-Zeit im zweiten Zyklus nach 20 ms. Die Ausführung der Task wird abgebrochen und die Systemtask "Watchdog" aufgerufen.

Im **Beispiel 2** ist die Watchdog-Zeit auf 20 ms eingestellt. Er spricht deshalb auf die Zeitüberschreitung der Task im zweiten Zyklus nicht an. Die Task setzt lediglich für den nächsten Zyklus aus und wird erst wieder nach 30 ms im vierten Zyklus ausgeführt.

9.6 Tasks einfügen und Programme zuweisen

Tasks einfügen

Um eine Task einzufügen, müssen in MULTIPROG folgende Schritte ausgeführt werden:

- Im Projektbaum, unter der Ressource für die jeweilige Steuerung, mit der rechten Maustaste auf den Ordner *Tasks* klicken, um das Kontextmenü zu öffnen (Bild 93).

Bild 93: Multi-Task-System, Einfügen einer Task in MULTIPROG

- Den Menüpunkt *Einfügen/Task wählen*, es erscheint der Dialog *Einfügen*.
- Den Namen für die Task eingeben.
- Im Listenfeld *Task-Typ* den gewünschten Task-Typ einstellen. Es kann zwischen einer Default-Task, zyklischen Task, der Ereignis- oder System-Task gewählt werden. Hinweis: Wenn der Tasktyp 'DEFAULT' nicht in dem Listenfeld aufgeführt ist, besitzt die Ressource bereits eine Default-Task.
- Den Dialog mit *OK* bestätigen.

Es erscheint der Dialog *Task-Einstellungen für ...* Abhängig von der zuvor ausgewählten Task enthält der Dialog verschiedene Text- und Listenfelder.

Für die jeweilige Task müssen folgende Parameter eingegeben werden:

Tabelle 41: Multi-Task-System, Taskparameter

Task	Parameter
zyklische Task	Zeitintervall
Ereignis-Task	Ereignisnummer (Nummer des Interrupt)
System-Task	Nummer eines Systemprogrammes

Bei der Vergabe der Priorität müssen unbedingt die Ausführungen im Abschnitt Task-Prioritäten beachtet werden.

Programme müssen Tasks zugewiesen werden, um sie auszuführen. Zuweisen eines Programmes zu einer Task bedeutet, dass eine Instanz des Programmes ausgeführt wird, wenn die Task aktiviert wird. Von einem Programm können verschiedene Instanzen verschiedenen Tasks zugeordnet werden.

Einer Task können mehrere Programme zugewiesen werden. In diesem Fall wird das erste Programm im Taskverzeichnis als erstes ausgeführt. Danach wird das Programm darunter ausgeführt usw.

Programme zuweisen

Um Programme zuzuweisen, müssen in MULTIPROG folgende Bedienschritte durchgeführt werden:

- Im Projektbaum mit der rechten Maustaste auf das Symbol der Task klicken, in die das Programm eingefügt werden soll (Bild 94).
- Im Kontextmenü *Einfügen* / *Programminstanz* wählen.
- Einen Instanznamen für das Programm in das Feld *Programminstanz* eingeben.
- Im Listenfeld *Programmtyp* das gewünschte Programm einstellen.
- Den Dialog mit OK bestätigen.

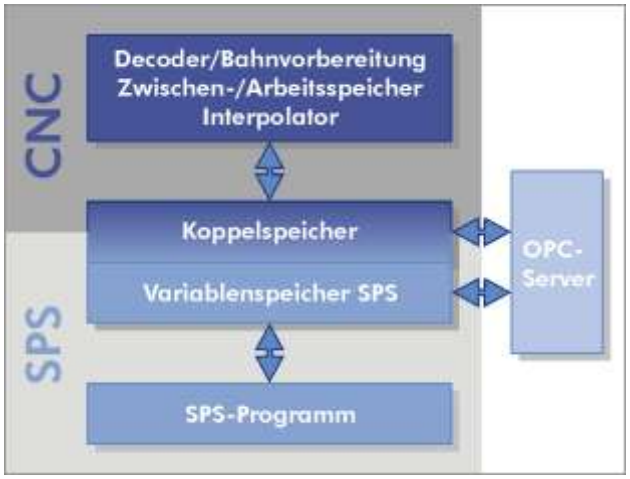
Das Programmsymbol wird in den Projektbaum eingefügt.

Bild 94: Multi-Task-System, Zuweisen von Programmen in MULTIPROG

10 Der Koppelspeicher

Der Koppelspeicher verbindet die Ablaufsteuerung der SPS und die Bewegungsfunktionen der CNC. Die beiden Steuerungssysteme arbeiten zum Datenaustausch synchron auf den Speicher, wobei die SPS eine Masterfunktion übernehmen kann. Auch Visualisierungssysteme sind über OPC in die Kommunikation eingebunden.

Bild 95: Koppelspeicher als Verbindung von SPS und CNC



Die enge Verbindung des SPS- mit dem CNC-System ermöglicht die Realisierung von komplexen Funktionsabläufen, die mit getrennten SPS- und CNC-Steuerungen undenkbar wären. Die klassische SPS-Schnittstelle ermöglicht SPS-Funktionen im NC-Programm, z.B. das Setzen und Abfragen von SPS-Merkern. Die Synchronisierung der SPS-Task mit der CNC-Lageregelung bietet darüberhinaus weitreichende Optionen:

- Es entstehen keine Wartezeiten und kein Kommunikations-Overhead
- Die SPS kann alle Aktionen der CNC überwachen
- Die CNC kann über die SPS Sollwerte ausgeben
- Die SPS kann der CNC Führungsgrößen im Lageregeltakt vorgeben

10.1 Variablen und Tasks

Die Koppelspeicherdaten sind Variablen gemäß IEC 61131-3, die bei der Projektierung im Programmiersystem MULTIPROG als globale Variablen deklariert werden. Sie sind standardmäßig dem OPC-Server zugänglich und werden z.B. im Bedientool Schleicher-Dialog angezeigt.

Im Multitask-Betriebssystem ist die SPS-Task 6 mit der Interpolationstask der CNC-Steuerung synchronisiert. Die Zykluszeit der Task 6 richtet sich dann nach dem IPO-Takt der CNC.

10.2 Zugriff auf den Koppelspeicher

Die Datenstruktur des Koppelspeichers wird bei der Installation des SPS- bzw. CNC-Betriebssystems angelegt. Bei reiner SPS sind nur die Variablenbereiche für SPS-Spezifika und Fehler (*plcSect* und *errSect*, siehe unten) angelegt. Das SPS-Programm hat Zugriff auf den gesamten Koppelspeicher über die globale Variable *plcMem* (bei SPS-Steuerungen) bzw. *cncMem* (bei CNC-Steuerungen). Die einzelnen Komponenten für den lesenden und schreibenden Zugriff (z.B. Versionsnummern, Fehlermeldungen, Bitsignale, Wortbereiche, NC-Daten, CAN-Daten, etc.) sind in Bereiche zusammengefasst. Die remanenten Variablen (Retain) belegen jeweils eigene Bereiche.

- SPS-spezifischer Bereich plcSect
- Fehlerbereich errSect
- Allgemeiner Bereich comSect
- Allgemeiner Bereich (Retain) comSect
- Systembereich sysSect
- Systembereich (Retain) sysSect
- Achsbereich axSect
- Achsbereich (Retain) axSect

Mit der Schreibweise *Globale_Variable.Bereich.Komponente* kann auf die einzelnen Bereiche und Komponenten des Koppelspeichers zugegriffen werden. Zum Beispiel kann das SPS-Programm die Versionsnummer der Betriebssoftware der Steuerung aus der Variablen `cncMem.plcSect.lOSVersion` auslesen. Zur Einbindung des Koppelspeichers in das Programmiersystem MULTIPROG (siehe Seite 46).

Visualisierungssysteme haben Zugriff auf den Koppelspeicher über die OPC-Schnittstelle. Da der OPC-Server strukturierte Variablen nicht darstellen kann, ist die gesamte Datenstruktur des Koppelspeichers als eindimensionale Liste abgebildet. Die Namen sind dabei aus zwei Teilen zusammengesetzt, die durch einen Unterstrich voneinander getrennt sind. Der vordere Namensteil stellt den Zugriffspfad dar, der hintere Namensteil ist identisch mit dem Komponentennamen der SPS-Variablen. Die Versionsnummer der Betriebssoftware könnte beispielsweise aus der OPC-Variablen `cmpS_lOSVersion` ausgelesen werden.

10.3 Hilfe zum Koppelspeicher

Eine genaue Beschreibung der Koppelspeicherstruktur und sämtlicher Variablen ist über die Online-Hilfe sowohl in MULTIPROG als auch im Schleicher-Dialog erreichbar.

10.4 Weitere Hintergrundinformationen zum Koppelspeicher

Beim Einfügen des Koppelspeichers in ein SPS-Projekt werden folgende Elemente hinzugefügt bzw. aktualisiert:

- Das Datentyp-Arbeitsblatt *SharedMemory_Types*; hier ist die Datenstruktur des Koppelspeichers deklariert.

  Das Datentyp-Arbeitsblatt wird ab der Koppelspeicher-Version 8 eingefügt. Bei früheren Versionen wird kein Datentyp-Arbeitsblatt eingefügt, da in dem Fall die Deklaration der Koppelspeicher-Struktur in der zugehörigen Anwender-Bibliothek *SchleicherLib_Vxxx* enthalten ist (Koppelspeicher-Version 7 in *SchleicherLib_V007*, Koppelspeicher-Version 6 in *SchleicherLib_V006* usw.).

- Die globalen Variablen `plcMem` (für SPS-Steuerungen) bzw. `cncMem` (für CNC-Steuerungen). Siehe Arbeitsblatt *Global_Variable*s, Gruppe *SharedMemory_Variables*.

  Diese Variablen repräsentieren den gesamten nichtremanenten (non retain) Bereich des Koppelspeichers. Über die Punktschreibweise kann das SPS-Programm wie oben erläutert auf die einzelnen Komponenten (Variablen) des Koppelspeichers zugreifen.

  Ab der Koppelspeicher-Version 8 existiert für CNC-Steuerungen neben dem nichtremanenten auch ein remanenter (retain) Bereich des Koppelspeichers. Die Werte der Variablen dieses Bereiches bleiben im Gegensatz zum nichtremanenten Bereich auch nach dem Ausschalten der Steuerung erhalten. Hierfür wird zusätzlich die globale Variable `cncRMem` eingefügt.

- Die globalen Variablen `cmpS..., cmeS..., cmcS..., cmsS..., cmaS...`. Siehe Arbeitsblatt *Global_Variable*s, Gruppen *PLC_Common*, *CNC_Common*, *CNC_System_x* (x steht für die Nummer des CNC-Teilsystems, für jedes Teilsystem existiert eine solche Gruppe mit systemspezifischen Variablen) und *CNC_Axis_y* (y steht für die Nummer der CNC-Achse, für jede Achse existiert eine solche Gruppe mit achsspezifischen Variablen).

  Diese Variablen sind für den Zugriff von Visualisierungssystemen oder ähnlichen Programmen auf den Koppelspeicher vorgesehen. Sie stellen den gesamten Koppelspeicher über die OPC-Schnittstelle zur Verfügung.

  Diese Variablen stellen den Koppelspeicher in unstrukturierter Form zur Verfügung. Es werden nur einfache Datentypen (BOOL, DINT, REAL, STRING) und Felder von einfachen Datentypen verwendet. Diese Vorgehensweise ist erforderlich, da strukturierte Daten (wie sie die Variablen `plcMem` und `cncMem` beinhalten) nicht über die OPC-Schnittstelle übertragen werden können.

11 Die CNC

Die XCA Steuerung ist eine CNC mit bis zu 64 Achsen/Spindeln und integrierter, leistungsfähiger SPS.

Funktionsübersicht

- Bis zu 32 Teilsystemen mit insgesamt 64 Achsen/Spindeln
- Technologien für Bohren, Fräsen, Schleifen, Handling
- Endlosdrehende Rundachsen
- Spindelpaket mit umfangreicher Funktionalität, wie z.B. Gewinde-schneidfunktionen, variable Pulsbewertung, orientierter Spindelhalt
- Synchronspindel
- Programmierbare Beschleunigung
- Elektronisches Getriebe
- 2D+n-Schraubenlinien-Interpolation
- Vorschub und Eilgang :
  0,001 mm/min bis 999 m/min
- Werkzeugradiuskorrektur mit An- und Abfahrstrategien
- Werkzeuglängenkorrektur
- Interpolatorische Spindelsteigungs- und Messsystemfehler-kompensation
- Losekompensation
- Arbeitsfeldbegrenzung
- Softwareendschalter

Die CNC-Programmierung der XCx wird ausführlich in einer eigenen Betriebsanleitung beschrieben (Seite 10).

12 Serielle Schnittstelle RS232

12.1 Inbetriebnahme der seriellen Verbindung über die RS232-Schnittstelle

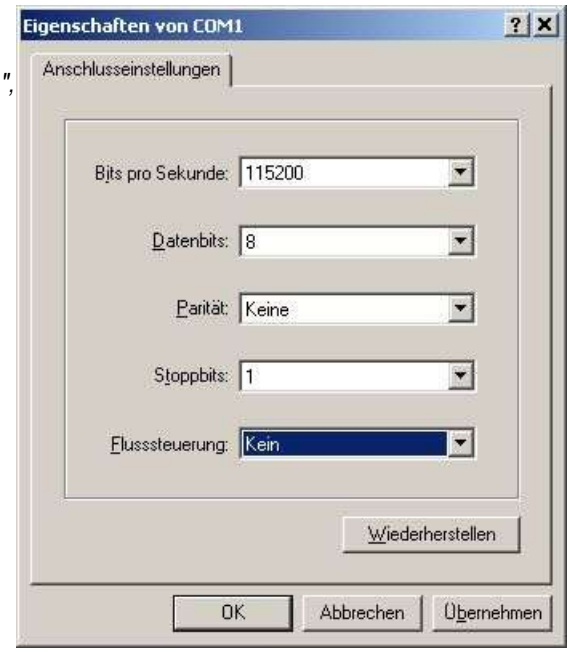
Die serielle Schnittstelle ist dem Echtzeitbetriebssystem zugeordnet und dient zum Anschluss von Bediengeräten und zur Systemdiagnose. Auf ihr erfolgt z.B. die Ausgabe des Bootprotokolls beim Hochfahren der XCx.

- Die Kabelverbindung zwischen dem Service-PC COM1 oder COM2 und dem XCx-Anschluss X12 herstellen (siehe Seite 19).
- Auf dem PC unter *Start* / *Programme* / *Zubehör* / *Kommunikation* das Programm *HyperTerminal* starten, einen Namen eingeben (z.B. XCx) und ein Symbol aussuchen.
- Im Dialogfenster *Eigenschaften von <Name>* unter *Verbindung herstellen über* die Direktverbindung über COM1 auswählen und dann auf den Button *Konfigurieren* klicken.

Bild 96: Dialogfenster "Eigenschaften von ...", Auswahl der Verbindung

- In *Eigenschaften von COM1* diese Parameter einstellen.
  - Bits pro Sekunde: 115200
  - Datenbits: 8
  - Parität: Keine
  - Stoppbits: 1
  - Flusssteuerung: Kein

Bild 97:
Dialogfenster "Eigenschaften von COM1", Anschluss-einstellungen



- Die XCx einschalten oder RESET auslösen.

Während des Steuerungsanlaufes wird im HyperTerminal-Dialogfenster das Bootprotokoll ausgegeben.

13 Weitere Betriebssoftware

13.1 Windows embedded

Neben den Echtzeit-Betriebssystemen für SPS und CNC ist die XCA Steuerung mit dem Betriebssystem Windows embedded ausgestattet, das zeitunkritische Aufgaben wie Visualisierung oder Diagnose in gewohnter Bedienumgebung ermöglicht (Seite 11).

Windows embedded ist ein Microsoft-Betriebssystem, das auf dem gleichen Quellcode wie Windows basiert. Das Betriebssystem ist vollständig modular aufgebaut und bietet damit dem Hersteller die Möglichkeit, es an die Erfordernisse eines Gerätes anzupassen.

13.2 Schleicher X-Manager

Der Schleicher X-Manager, Version 1.x für XCA 1100 und Version 2.x für XCA 1200, ermöglicht die Konfiguration von wichtigen Parametern und Einstellungen zum Betrieb der XCA Steuerungssysteme wie:

"IP Configuration", "EWF Configuration", "NFS and Hardlink" (anwendbar mit XCA 1100 und XCA 1200)

"RTOS Settings" (anwendbar mit XCA 1200)

13.2.1 "IP Configuration"

Hierüber lassen sich die für den Betrieb erforderlichen Netzwerk-Einstellungen an die Gegebenheiten des lokalen Netzwerks (Hausnetz) anpassen (siehe Kap. Inbetriebnahme der Netzwerk-Schnittstelle Seite 33).

Auf der XCA Steuerung läuft "gleichzeitig" das Echtzeitbetriebssystem VxWorks sowie Windows embedded. Beide Betriebssysteme kommunizieren über einen gemeinsamen Speicher (Shared Memory) und ein virtuelles Netzwerk miteinander. Hierzu werden auf beiden Seiten sog. "virtuelle Netzwerkadapter" ("RtOS Virtual Network") zur Verfügung gestellt.

Die Netzwerkschnittstelle wird in folgenden Schritten in Betrieb genommen:

- Vergabe einer **Hausnetz-Adresse** zur Anbindung der Steuerung an ein vorhandenes Hausnetz (z.B. 10.208.3.212)
- Ändern der Windows-seitigen Adresse des "RtOS Virtual Network". Nach Möglichkeit sollte das Default-Schema (z.B. 192.168.xxx.1) beibehalten werden!
- Ändern der VxWorks-seitigen Adresse des "RtOS Virtual Network". Nach Möglichkeit sollte das Default-Schema (z.B. 192.168.xxx.2) beibehalten werden!
- Ändern der Netzwerkeinstellungen (Routing) am Programmier-PC zur Kommunikation mit der Steuerung über das Hausnetz (z.B. zur Programmierung mit MULTIPROG)

Wichtig!

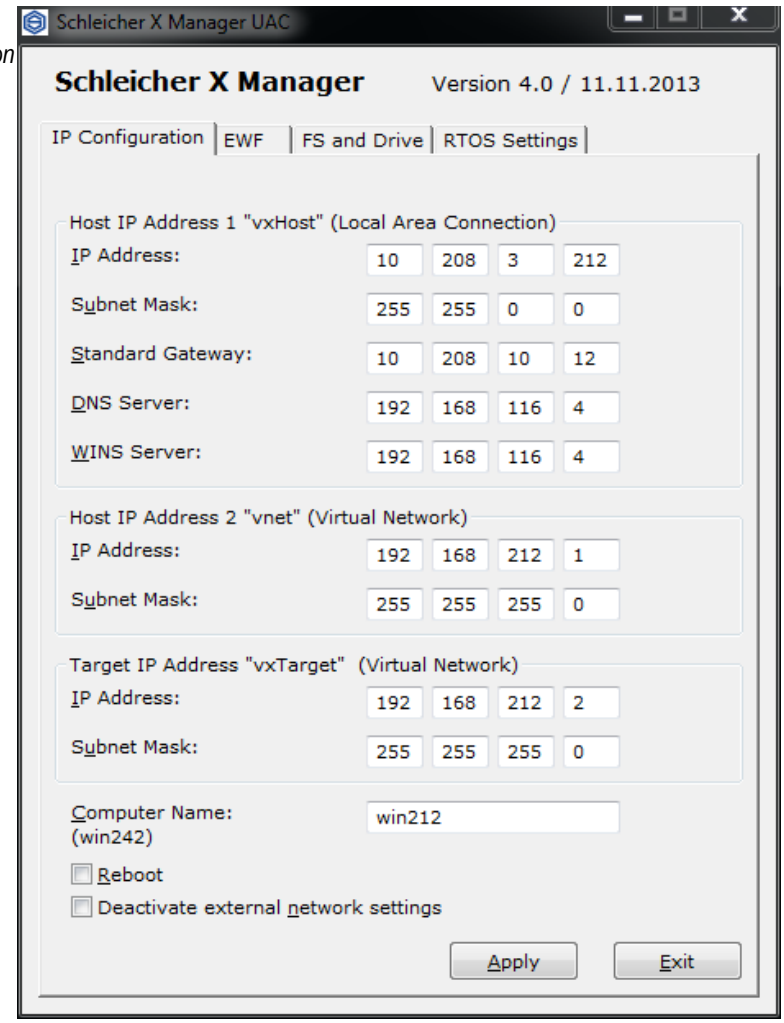
Für die Installation mit Anbindung an ein vorhandenes Hausnetz ist vor dem Anpassen der IP-Adressen die Vorgehensweise mit dem Netzwerkadministrator für das jeweilige Hausnetz zu klären.

1. Host IP Address 1 "vxHost" (Local Area Connection)
   - IP-Adresse und Maske der XCA Steuerung Steuerung im Hausnetz
     (in der Datei c:\windows\system32\host wird hierfür der Aliasname vxHost definiert)
   - IP-Adresse des Standard Gateways im Hausnetz
   - IP-Adresse des Standard DNS im Hausnetz
   - IP-Adresse des Standard WINS im Hausnetz (optional)
2. Host IP Address 2 (RtOS Virtual Network)
   - IP-Adresse und Maske des "RtOS Virtual Network" der XCA Steuerung (Windows-Seite, s. Abb.)
3. Target IP Address #1 (alias vxTarget)
   - IP-Adresse und Maske des "RtOS Virtual Network" der XCA Steuerung (VxWorks-Seite). (in der Datei c:\windows\system32\host wird hierfür der Aliasname vxTarget definiert)
4. "Computer Name" (s.Tooltips)
   - Der "Computername" muss netzwerkweit eindeutig sein und darf nur einmal vergeben werden (z.B. win212)
   - In Klammern wird der aktuell verwendete Computername angezeigt.
   - Die Änderung des Computernamens erfodert einen Neustart der Steuerung
5. "Deactivate external network settings" (s.Tooltips)
   - Deaktiviert die Gateway-, DNS- und WINS Einstellungen. Hiermit kann die Zeit bis SPS-Start, z.B. wenn an der Ethernet-Schnittstelle der XCx1100 kein Netzwerkkabel zum LAN angeschlossen wurde, dramatisch verkürzen werden.
     Hintergrund:
     Die XCA Steuerung stellt einen integrierten Switch zum Anschluss mehrerer Ethernet-Geräte bzw. des Hausnetztes bereit. Damit ist es jedoch Windows nicht mehr möglich zu unterscheiden, ob sich die Steuerung tatsächlich am Hausnetz (mit Zugriff auf DNS, WINS etc.) befindet oder nicht. Um Konflikte ("*Race-Conditions*") beim Starten des Netzwerk-Stacks und des Echtzeitbetriebs-systems zu vermeiden, wird daher zuerst gewartet, bis das Netzwerk vollständig funktionsbereit ist oder die entsprechenden Netzwerk-Timeouts abgelaufen sind. Diese in der Praxis recht langen Wartezeiten sind in dem Fall, dass kein Netzwerk angeschlossen wurde, jedoch unnötig.
6. Reboot (s.Tooltips)
   - standardmäßig ist der EWF auf dem System aktiviert, d.h. alle Änderungen an der Systempartition sind nach einem Neustart verloren
   - um die veränderten IP-Einstellungen dauerhaft zu übernehmen, wird daher automatisch ein "Commit Changes" Auftrag abgesetzt
   - alle Änderungen werden jedoch erst durch einen Neustart der Steuerung übernommen (d.h. nach dem regulären Herunterfahren von Windows)
   - Bei aktivierter Reboot-Checkbox erfolgt der Neustart automatisch.

Hinweise!

- **Für einige Einstellungen sind gesonderte Tooltip-Hilfen verfügbar.**
- **Für "Computer Name" wird in den Tooltips ein "Routing"-Befehl für einen "Programmier-PC" ausgegeben. Je nach Hausnetz-Konfiguration sind gegebenenfalls zusätzliche Einstellungen erforderlich.**
- **Mit "F2" kann die Default IP Konfiguration (Auslieferungs-zustand) bzw. mit "F3" die aktuelle IP Konfiguration in die Eingabemasken ("IP Configuration") geladen werden**
- **Zur Unterstüzung des vorgeschlagenen IP Adress-Schemas (s.o.) dient Hotkey "F5", mit "F5" wird die letzte Nummer der Host IP Adresse (z.B. 212) automatisch für die Vergabe der Host IP #2 bzw. der Target IP-Adresse herangezogen**

Bild 98:
X Manager, IPConfiguration

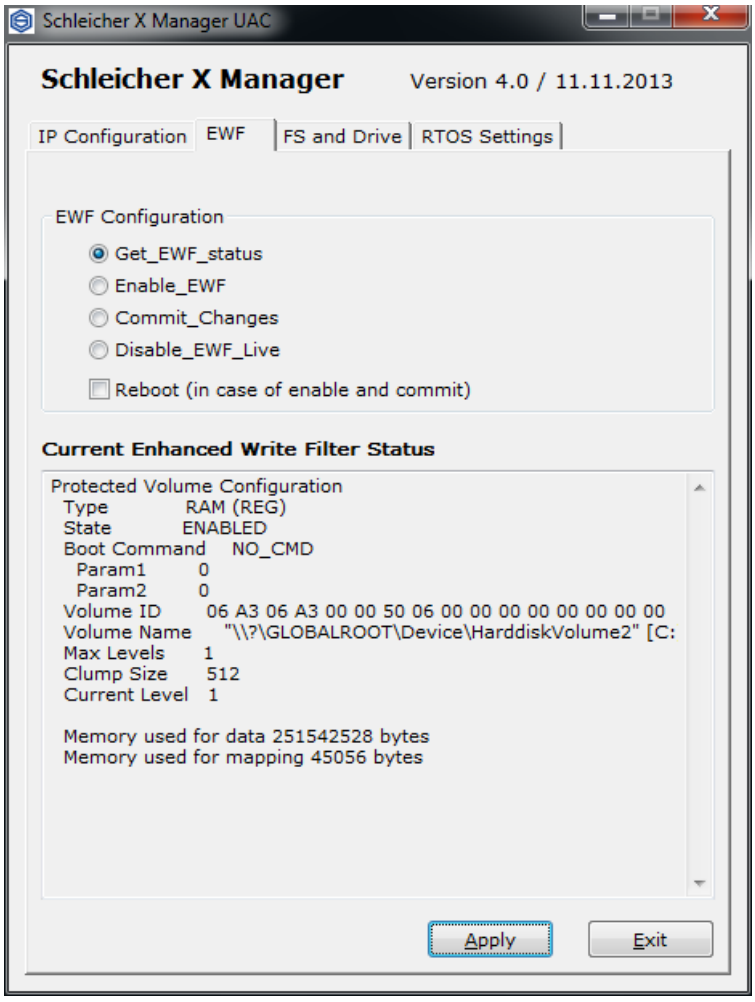

Schleicher X Manager UAC
Schleicher X Manager
Version 4.0 / 11.11.2013
IP Configuration
EWF
FS and Drive
RTOS Settings
Host IP Address 1 "vxHost" (Local Area Connection)
IP Address: 10 208 3 212
Subnet Mask: 255 255 0 0
Standard Gateway: 10 208 10 12
DNS Server: 192 168 116 4
WINS Server: 192 168 116 4
Host IP Address 2 "vnet" (Virtual Network)
IP Address: 192 168 212 1
Subnet Mask: 255 255 255 0
Target IP Address "vxTarget" (Virtual Network)
IP Address: 192 168 212 2
Subnet Mask: 255 255 255 0
Computer Name: (win242)
win212
Reboot
Deactivate external network settings
Apply
Exit

13.2.2 "EWF Configuration"

Der Enhanced Write Filter ist eine Komponente im PC-Betriebssystem Windows embedded der XCA Steuerung. Sie dient zum Schutz einer oder mehrerer Partitionen (Volumes, z.B. auf einem CF-Drive) vor Veränderungen. Schreibende Zugriffe werden bei aktiviertem EWF in ein sogenanntes Overlay (z.B. im System-RAM) umgeleitet. Nach dem Ausschalten des Systems sind diese verloren. Um dennoch gewünschte Veränderungen am System vornehmen zu können, kann der Inhalt des Overlay-Speichers über entsprechende Systemkommandos gezielt zurückgeschrieben oder der EWF temporär deaktiviert werden.

- "Get EWF status"
  gibt die aktuelle Konfiguration und den Status des EWF aus
- "Enable EWF"
  aktiviert den EWF
- "Commit changes"
  schreibt den Inhalt des Overlay-Speichers zurück
- "Disable EWF Live"
  deaktiviert den EWF
- "Reboot"
  Eine Zustandsänderung des EWF (mit Ausnahme der Deaktivierung) wird lediglich beauftragt und erst mit einem System-Reboot tatsächlich ausgeführt. Der Neustart kann mit der gesetzten Option "Reboot" in Verbindung mit dem entsprechenden EWF-Kommando sofort ausgelöst werden.

Bild 99:
X Manager, EWF Configuration

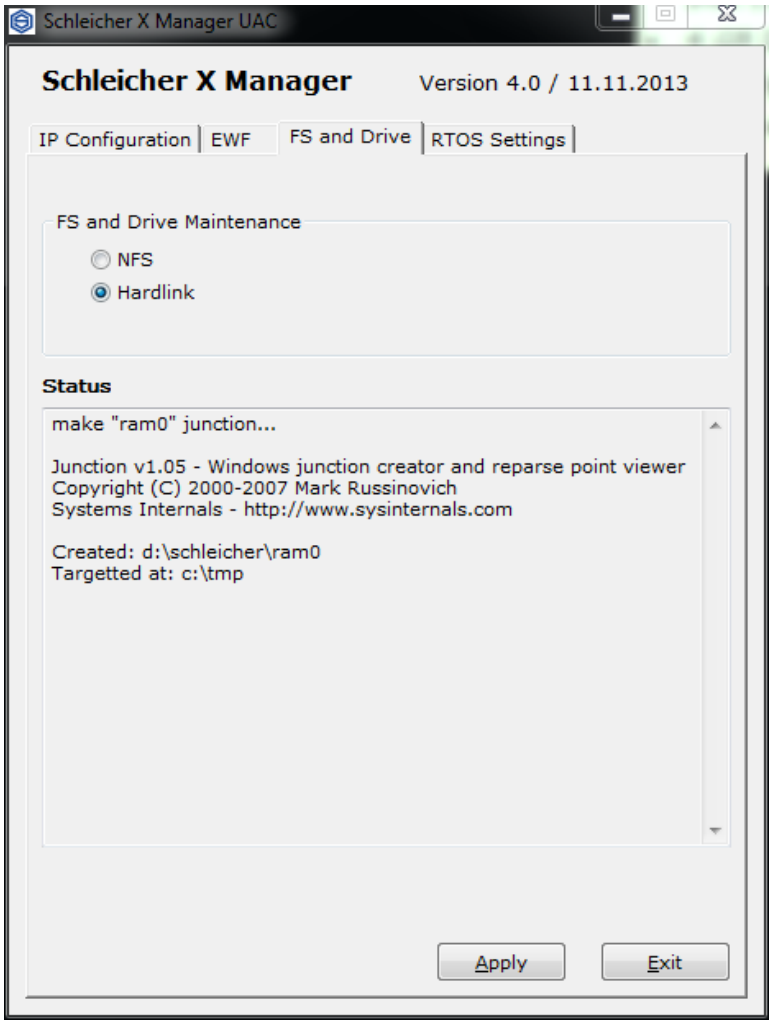


"NFS and Hardlink"

Nach dem Restore eines Windows embedded Images (z.B. bei vom Ausgangssystem abweichenden CF-Cards) kann Windows u.U. die zum Betrieb des Echtzeitbetriebssystems erforderlichen NFS-Freigaben ("Network File System") "verlieren".

- "Get_NFS_status"
  gibt die aktuelle NFS Konfiguration aus
- Refresh_NFS
  setzt alle erforderlichen NFS-Freigaben erneut:
  c:\tmp = "ram0" virtuelle RAM-Disk, Ablage für temporäre Dateien, nach jedem Abschalten
  d:\schleicher = "pc" Basisverzeichnis für VxWorks
- Refresh_Hardlink
  stellt den u.U. aufgehobenen NTFS-Hardlink von d:\schleicher\ram0 auf c:\tmp wieder her

Bild 100:
X Manager, NFS and Hardlink

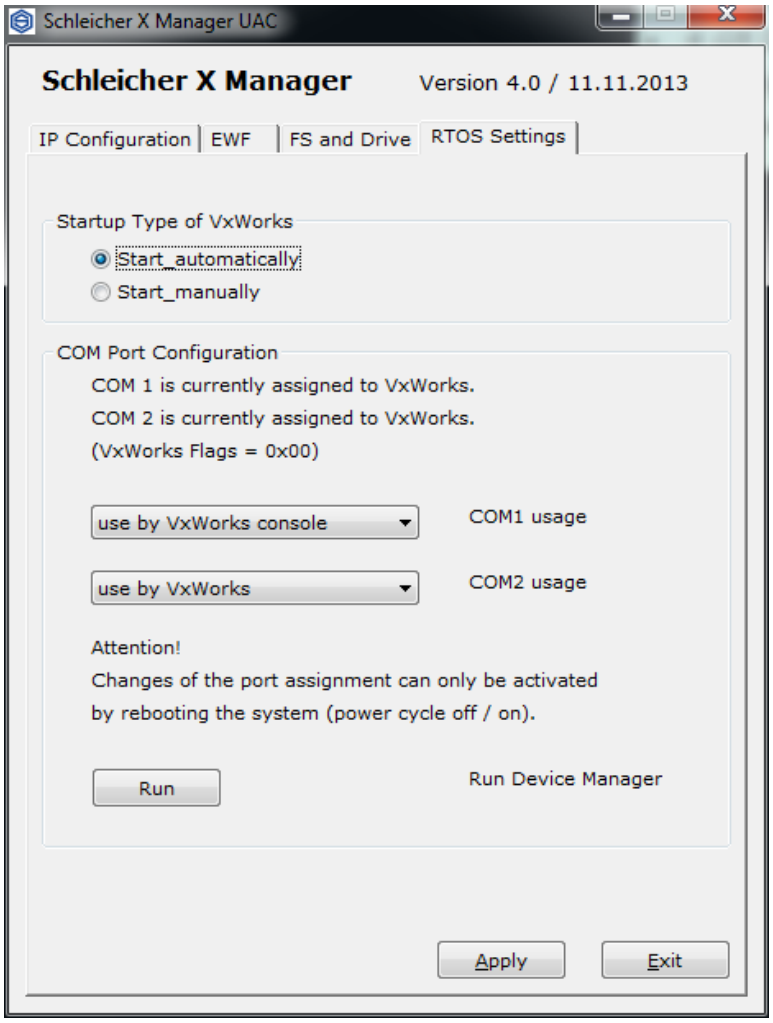


13.2.3 "RTOS Settings"

Nur für XCA1200

- "Startup Type of VxWorks"
  - Beim Start von Windows wird VxWorks automatisch gestartet
  - VxWorks muss manuell gestartet werden (mit Hilfe des MFA-Utilities)
- "COM Port Configuration"
  - Legt die Zuordnung der COM-Ports zum jeweiligen Betriebssystem fest (Windows oder VxWorks)
  - für COM1 kann zudem festgelegt werden, ob eine Console aktiviert wird oder nicht
- "Run"
  - führt den Gerätemanager aus (zur Kontrolle der korrekten Zuordnung durch den Hypervisor)

Bild 101:
X Manager, RTOS Settings



13.3 Remote Desktop UltraVNC

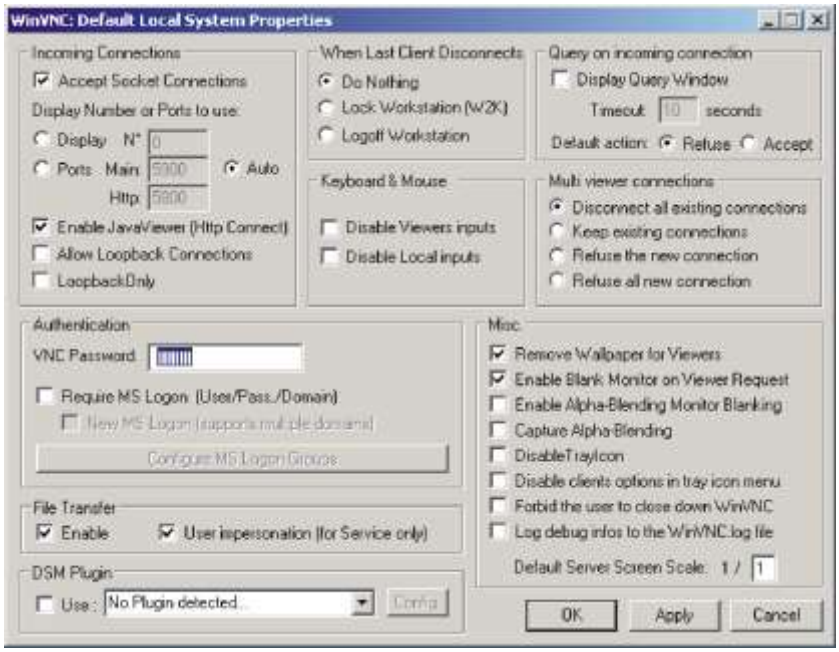
UltraVNC ist eine Fernwartungs-Software unter Windows, die von einem PC aus den Remote-Zugriff über Netzwerk oder Internet auf den Desktop eines entfernten Windows-Rechners ermöglicht.

Die Software arbeitet nach dem Client-Server-Modell, wobei der Server auf dem zu überwachenden Rechner läuft (im Falle der XCx also unter Windows embedded). Der Client empfängt die Bildschirmausgaben des Servers und schickt diesem seinerseits Maus- und Tastatureingaben.

Für die Verbindung PC <-> XCA 1100 ist das Password in der Defaulteinstellung "pp" (siehe Kommunikation zur Programmierung mit MULTIPROG (Schritt 4)

in Bild 21:
Eingeben der IP-Adresse
in MULTIPROG).

Bild 102: Fernwartungssoftware UltraVNC



UltraVNC basiert auf dem Netzwerk-Protokoll VNC (Virtual Network Computing) und steht als Open-Source-Variante unter der GNU General Public License zum freien Download zur Verfügung. Das Programm ist lauffähig unter Windows 95/98/NT/2000/XP/Vista/Windows 7 und 8. Es bietet Funktionen wie Datenverschlüsselung, Passwortabfrage, Mirror Video Driver (für den lesenden Zugriff auf den Remote-Desktop), Datei- und Verzeichnistransfer sowie einen Text-Chat.

Der Server von UltraVNC ist auf der XCx-Steuerung vorinstalliert. Um den Client auf einem Wartungsrechner zu installieren, laden Sie die aktuelle Version von UltraVNC von der Webseite des Herstellers: http://www.uvnc.com. Auf der Seite finden Sie zahlreiche Hilfen zu Einrichtung und Bedienung, (Online-)FAQs, Foren und Tutorials.

14 Anhang

14.1 Technische Daten aller Module

Klimatische Bedingungen	
Betriebsumgebungstemperatur	0 ... +55°C (Kl. KV nach DIN 40040), senkrechter Einbau, freie Luftzirkulation
Lagertemperatur	-25 ... +70°C (Kl. HS nach DIN 40040)
Relative Luftfeuchte	10 ... 95% (Kl. F nach DIN 40040), keine Betauung
Luftdruck im Betrieb	860 ... 1060 hPa

Mechanische Festigkeit	
Schwingen	nach DIN EN 60068-2-6 10 ... 57 Hz konstante Amplitude 0,075mm 57 ... 150 Hz konstante Beschleunigung 1 g
Schocken	nach DIN EN 60068-2-27, Sinus-Halbwelle 15g / 11ms
Freier Fall	nach DIN EN 60068-2-32, Fallhöhe 1m (mit Originalverpackung)

Elektrische Sicherheit	
Schutzklasse	Klasse I nach EN 60536 (Basisisolierung und Schutzleiteranschluss)
Schutzart	IP 00 nach EN 60529
Luft-/Kriechstrecken	nach DIN EN 61131-2, zwischen Stromkreisen und Körper sowie zwischen galvanisch getrennten Stromkreisen, entsprechend Überspannungskategorie II, Verschmutzungsgrad 2
Prüfspannung	AC 350 V/50Hz für Geräte-Nennspannung DC 24V AC 1350 V/50Hz für Geräte-Nennspannung AC 230V

Elektromagnetische Verträglichkeit	
Elektrostatische Entladung	nach EN 61000-4-2, 8 KV Luftentladung, 4 KV Kontaktentladung
Elektromagnetische Felder	nach EN 61000-4-3, Feldstärke 10 V/m, 80 ... 1000 MHz
Schnelle Transienten (Burst)	nach EN 61000-4-4, 2 KV auf AC/DC-Versorgungsleitungen, 1 KV auf E/A-Signalleitungen
Störaussendung	nach EN 55011, Grenzwertklasse A, Gruppe 1

14.2 Zubehör und Software

Bezeichnung	Beschreibung	Artikelnummer
MULTIPROG 4.x	SPS Programmiersystem nach IEC61131-3	R4.320.0640.0
Service Pack	Steuerungssoftware für alle Schleicher-Steuerungen, AddOns, Schleicher-Dialog, Dokumentationen und Service-Informationen	R4.320.0590.0
ProCANopen	Netzwerk-Konfigurationssoftware	R4.320.0500.0
CANcardY	1-fach CANopen-Interface, PCMCIA-Steckkarte	R4.321.0020.0

Tabelle 42: Zubehör und Ersatzteile

14.3 Warenzeichenvermerke

- WINDOWS ist ein eingetragenes Warenzeichen der Microsoft Corporation.
- CANopen ist ein eingetragenes Warenzeichen von CAN in Automation e.V,
- ProCANopen ist ein eingetragenes Warenzeichen von Vector Informatik GmbH.
- VxWorks ist ein eingetragenes Warenzeichen der Wind River Systems Inc.
- PROFIBUS ist ein eingetragenes Warenzeichen der PROFIBUS Nutzerorganisation.
- MULTIPROG ist ein eingetragenes Warenzeichen der KW-Software GmbH

Alle anderen Warenzeichen oder Produktnamen sind eingetragene Warenzeichen der jeweiligen Firmen.

14.4 Abbildungsverzeichnis

Bild 1: Systemübersicht XCA Steuerungen , Erweiterungsmodule und Peripherie 12
Bild 2: Aufbau des Gesamtsystems 13
Bild 3: Steckplatz der Steuerung auf dem Baugruppenträger 14
Bild 4: Montage der Module auf dem Baugruppenträger 14
Bild 5: XCA Steuereinheit, CPU und Kühlkörper 15
Bild 6: XCA Steuereinheit, Schnittstellen, Bedienelemente und LED-Anzeigen 16
Bild 7: Installation von MULTIPROG 30
Bild 8: Installation der AddOns 31
Bild 9: Installation des Schleicher-Dialogs 32
Bild 10: Anschluss der Ein-/Ausgabegeräte an die XCA Steuerung 33
Bild 11: Inbetriebnahme der Netzwerkschnittstellen in vier Schritten 34
Bild 12: Dialogfenster "Systemeigenschaften", Eingabe der Computerbeschreibung 35
Bild 13: Dialogfenster "Computernamen ändern" 35
Bild 14: Auswahl der Netzwerkverbindungen 36
Bild 15: Eigenschaften von Local Area Connection 37
Bild 16: Eintragen von IP-Adresse und Subnetzmaske für Local Area Connection 37
Bild 17: Eigenschaften von RtOS Virtual Network 38
Bild 18: Eintragen von IP-Adresse und Subnetzmaske für RtOS Virtual Network 38
Bild 19: Einrichten der Netzwerkroute zur Programmierung mit MULTIPROG 39
Bild 20: Aufruf der Ressource-Einstellungen in MULTIPROG 40
Bild 21: Eingeben der IP-Adresse in MULTIPROG 40
Bild 22: Aufruf der Verbindung zur XCA Steuerung in MULTIPROG 41
Bild 23: Ressource-Fenster bei Online-Verbindung zur XCA Steuerung 41
Bild 24: Neues Projekt in MULTIPROG öffnen 42
Bild 25: Steuerungstyp für neues Projekt auswählen 42
Bild 26: Beispiel für Projektbaum in MULTIPROG 43
Bild 27: Speicherdialog aufrufen 43
Bild 28: Dialogfenster "Projekt speichern" 43
Bild 29: Projekt kompilieren 44
Bild 30: Projekt auf die Steuerung übertragen 44
Bild 31: Online-Anzeige der Variablen 45
Bild 32: Aufruf des Dialogfensters "SPS/MC/CNC Koppelspeicher" 46
Bild 33: Dialogfenster "SPS/CNC/MC Koppelspeicher" 47
Bild 34: Auslesen der Betriebsoftware-Version 48
Bild 35: Darstellung der Koppelspeicher-Fehlermeldung im Schleicher-Dialog 49
Bild 36: Dialogfenster "XUIO-Konfiguration" 50
Bild 37: Menüeintrag "Extras / Schleicher Add-Ons" 52
Bild 38: Menüeintrag "Extras / XIO Konfiguration" 52
Bild 39: Dialogfenster "XRIO-Konfiguration / XUIO-Konfiguration" 53
Bild 40: Dialogfenster "XUIO-Konfiguration" 54
Bild 41: Knoten "IO_Configuration" 55
Bild 42: Dialogfenster "I/O-Konfiguration" 55
Bild 43: Knoten "Global_Variables" 56
Bild 44: Einfügen der globalen Variablen in das SPS-Projekt 56
Bild 45: Dialogfenster "XUIO-Konfiguration", Modul hinzufügen 57
Bild 46: Dialogfenster "XUIO-Konfiguration", Eigenschaften 58
Bild 47: Dialogfenster "Modul-Eigenschaften", Einstellen der Optionen 58
Bild 48: Anlegen der Variablen für die Interruptverarbeitung 59
Bild 49: Knoten für Interrupt-freigabe und -verarbeitung 61
Bild 50: Globale Variablen für Interruptzählung und Kommunikation 61
Bild 51: Zyklische Task "XUIO" mit POE "UBE32_Interrupt" 63
Bild 52: Einstellungen für zyklische Task 63
Bild 53: Anlegen der Event-Tasks für die Interrupts 63
Bild 54: Einstellungen für Event-Tasks 64
Bild 55: Einstellungen für Event-Tasks 64
Bild 56: Anschlussprinzip für CANopen-Netzwerk 66
Bild 57: DIP-Schalter am Kompaktmodul RIO 8 I/O CANopen 67
Bild 58: CANopen-Konfiguration, Dialogfenster "I/O-Konfiguration" 67
Bild 59: CANopen-Konfiguration, Dialogfenster "Eigenschaften" 68
Bild 60: CANopen-Konfiguration, Treiberparameter einstellen 68
Bild 61: CANopen-Konfiguration, Arbeitsblatt "Network_Variables" 69
Bild 62: CANopen-Konfiguration, Arbeitsblatt "I/O_Variables" 69
Bild 63: Aufruf der "Schleicher CANopen Konfiguration" in MULTIPROG 70
Bild 64: Schleicher CANopen Konfiguration 70
Bild 65: Aufruf von "ProCANopen Konfigurationswerkzeug" in MULTIPROG 73

Bild 66: Auswahl der CAN-Karte und Starten von ProCANopen ... 73
Bild 67: ProCANopen mit aktuellem CANopen-Projekt ... 73
Bild 68: Einlesen des Netzes mit ProCANopen ... 74
Bild 69: Einstellen der Scan-Optionen ... 74
Bild 70: Graphische Verknüpfung der Netzwerk-Knoten ... 75
Bild 71: Kommunikations beziehung zwischen Steuerung und I/O-Modul ... 75
Bild 72: Auswahl des Konfigurationsmanagers ... 76
Bild 73: Speichern der CAN-Konfiguration in den Konfigurationsmanager ... 76
Bild 74: Test der Netzwerkverbindung mit POE "CANdemo" ... 77
Bild 75: POE CANdemo mit dazugehörigem Variablen-Arbeitsblatt ... 77
Bild 76: Deklaration von PDD-Variablen in MULTIPROG ... 78
Bild 77: Taskleiste mit Schleicher-Logo ... 80
Bild 78: Fensteraufbau der MFA ... 80
Bild 79: MFA, Menü "Control System" ... 81
Bild 80: MFA, Menü "Startup Mode" ... 81
Bild 81: MFA, Menü "PLC" ... 82
Bild 82: MFA, Menü "Extras" ... 82
Bild 83: MFA, Menü "?" ... 83
Bild 84: Schleicher-Dialog, Startfenster ... 85
Bild 85: Schleicher-Dialog, Status- und Meldebereich ... 85
Bild 86: MULTIPROG, Wechseln der Betriebszustände ... 91
Bild 87: MULTIPROG, Kontextmenü "Bibliotheken" ... 94
Bild 88: MULTIPROG, Dialogfenster "Bibliothek einbinden" ... 95
Bild 89: MULTIPROG, Einbinden der Bibliothek "Serial" ... 95
Bild 90: Multi-Task-System, Prioritätsstufen ... 105
Bild 91: Multi-Task-System, Variablendeklaration ... 112
Bild 92: Multi-Task-System, Beispiel für Tasks und Watchdogs ... 114
Bild 93: Multi-Task-System, Einfügen einer Task in MULTIPROG ... 115
Bild 94: Multi-Task-System, Zuweisen von Programmen in MULTIPROG ... 116
Bild 95: Koppelspeicher als Verbindung von SPS und CNC ... 117
Bild 96: Dialogfenster "Eigenschaften von ...", Auswahl der Verbindung ... 121
Bild 97: Dialogfenster "Eigenschaften von COM1", Anschluss-einstellungen ... 122
Bild 98: X Manager, IPConfiguration ... 125
Bild 99: X Manager, EWF Configuration ... 126
Bild 100: X Manager, NFS and Hardlink ... 127
Bild 101: X Manager, RTOS Settings ... 128
Bild 102: Fernwartungssoftware UltraVNC ... 129

14.5 Tabellenverzeichnis

Tabelle 1: Zusätzliche Betriebsanleitungen 10
Tabelle 2: Pinbelegung der Ethernet-Schnittstellen X1, X2, X3 (RJ 45) 17
Tabelle 3: Pinbelegung der sercos III-Schnittstellen X4, X5 17
Tabelle 4: Pinbelegung der USB-Schnittstellen X6/7, X8/9 17
Tabelle 5: Pinbelegung der DVI-Schnittstelle X10 18
Tabelle 6: Pinbelegung der CAN-Schnittstellen X11 19
Tabelle 7: Pinbelegung der RS 232 / 422 / 485-Schnittstellen X12 19
Tabelle 8: LED-Anzeigen auf der Steuerung 20
Tabelle 9: LED-Anzeigen an den Ethernet- und sercos-III-Buchsen (5x RJ45) 21
Tabelle 10: Betriebsartenschalter 22
Tabelle 11: Reset-Taster 22
Tabelle 12: Liste der verfügbaren Baugruppenträger, Netzteile und Erweiterungsmodule für XCx und Promodul-U 27
Tabelle 13: Inhalt der CD-ROM 28
Tabelle 14: Systemvoraussetzungen 29
Tabelle 15: Wichtige Variablen des UBE 32 0,1l für die Interruptverarbeitung 60
Tabelle 16: Pinbelegung der verwendeten Stecker 66
Tabelle 17: Einstellen von Knotennummer und Datenübertragungsrate 67
Tabelle 18: Schleicher-Dialog, Aufteilung der Bedienoberfläche 85
Tabelle 19: Schleicher-Dialog, Status- und Meldebereich 85
Tabelle 20: SPS-Betriebszustände 90
Tabelle 21: MULTIPROG, Starten der Programmausführung 91
Tabelle 22: MULTIPROG, Stoppen der Programmausführung 92
Tabelle 23: MULTIPROG, Allgemeines Reset 92
Tabelle 24: Betriebsartenschalter 92
Tabelle 25: Systemvariablen 93
Tabelle 26: Bibliotheken und Funktionsbausteine in MULTIPROG 94
Tabelle 27: Bibliothek CANopen_Vxxx 96
Tabelle 28: Bibliothek CFB_Vxxx 96
Tabelle 29: Bibliothek CNC_Vxxx 97
Tabelle 30: Bibliothek Date_Time 98
Tabelle 31: Bibliothek MC_Vxxx 98
Tabelle 32: Bibliothek MMI 99
Tabelle 33: Bibliothek PLC_Vxxx 99
Tabelle 34: Bibliothek PNS_Vxxx 100
Tabelle 35: Bibliothek Profibus_Vxxx 100
Tabelle 36: Bibliothek Serial 100
Tabelle 37: Bibliothek XCx7_Vxxx 101
Tabelle 38: Multi-Task-System, Ereignis-Tasks 107
Tabelle 39: Multi-Task-System, System-Tasks 109
Tabelle 40: Multi-Task-System, Taskprioritäten 113
Tabelle 41: Multi-Task-System, Taskparameter 115
Tabelle 42: Zubehör und Ersatzteile 131

14.6 Index

Abbildungsverzeichnis ... 134
Abgesicherter Modus ... 81
Active-Error-Buffer ... 56, 91
AddOns, Installation ... 31
Anwenderbibliotheken ... *Siehe* Bibliotheken
Anwender-Task ... 108
Anwender-Task-Information ... 113
Automatik ... 88
Backplane ... 13
Baugruppenträger ... 26
BETRIEB Betriebszustand ... 92
Betriebsartenschalter ... 22, 94
Betriebszustände der SPS ... 92
Bibliothek
  CANopen_Vxxx ... 98
  CFB_Vxxx ... 98
  CNC_Vxxx ... 99
  Date_Time ... 100
  MC_Vxxx ... 100
  MMI ... 101
  PLC_Vxxx ... 101
  PNS_Vxxx ... 102
  Profibus_VxxxSerial ... 102
  SchleicherLib_Vxxx ... 102
  Serial ... 102
  XCx7_Vxxx ... 103
CANopen
  Anschluss an die Steuereinheit ... 19
  Anschlussprinzip ... 68
  Configuration manager ... 73
  Deklaration des I/O-Treibers ... 69
  EDS-Datei ... 73
  Einstellungen am RIO-Modul ... 69
  Feldbuskarte ... 73
  Konfiguration mit ProCANopen ... 73
  NMT manager ... 73
  Schleicher CANopen Konfiguration ... 72
  Spezifikationen ... 67
CAN-Schnittstellen ... 19
CNC ... 122
Darstellungskonventionen ... 6
Default-Task ... 112
Deklaration von Netzwerkvariablen ... 71
DVI-Schnittstelle ... 18
Echtzeit-Uhr ... 100
EDS-Datei ... 73
EIN Betriebszustand ... 92
Einbau des Automatisierungssystemes ... 13
Ereignis-Task ... 109
Error-Log-Book ... 56
Erweiterungsmodule ... 26
Ethernet
  Schnittstellen ... 17
  TCP/IP-Verbindung zur Programmierung ... 28
EWF Configuration ... 128
Fehlermeldungen ... 21
  Beispiel ... 56
Fernwartung ... 131
Firmware-Task Prioritäten ... 115
Fremdsoftware starten ... 90
Funktionsbaustein
  CLEAR_ERROR ... 101
  CLOSE_PROFILE ... 101
  CO_NET_CAN_SYNC ... 98
  CO_NET_GET_KERNEL_STATUS ... 98
CO_NET_GET_LOCAL_NODE_ID ... 98
CO_NET_GET_STATE ... 98
CO_NET_NMT ... 98
CO_NET_PING ... 98
CO_NET_RECV_EMY ... 98
CO_NET_RECV_EMY_DEV ... 98
CO_NET_RECV_ERR ... 98
CO_NET_RECV_ERR_DEV ... 98
CO_NET_RESTART_ALL ... 98
CO_NET_RESTART_CAN ... 98
CO_NET_SDO_READ ... 98
CO_NET_SDO_WRITE ... 98
CO_NET_SENDL2 ... 98
CO_NET_SHUTDOWN ... 98
CONNECT_V ... 98
DP_NET_GET_MSG ... 102
DP_NET_GET_STATE ... 102
DP_NET_PUT_MSG ... 102
FLUSH_PROFILE ... 101
GET_DATE ... 100
GET_MTS ... 101
GET_PROFILE_INT ... 101
GET_PROFILE_REAL ... 101
GET_PROFILE_STRING ... 101
GET_TIME ... 100
IBSM ... 103
MC_ANALOG ... 99
MC_ANALOG_1_AXIS ... 99
MC_CAN ... 99
MC_DP ... 99
MC_DP_1_AXIS ... 99
MC_GetCncAxis ... 100
MC_Home ... 100
MC_MoveAbsolute ... 100
MC_MoveAdditive ... 100
MC_MoveRelative ... 100
MC_MoveVelocity ... 100
MC_Power ... 100
MC_ReadActualPosition ... 100
MC_ReadAxisError ... 100
MC_ReadBoolParameter ... 100
MC_ReadParameter ... 100
MC_ReadStatus ... 100
MC_ReleaseCncAxis ... 100
MC_Reset ... 100
MC_Stop ... 100
MC_WriteBoolParameter ... 100
MC_WriteParameter ... 100
NEW_PROFILE ... 101
OPEN_PROFILE ... 101
PNSReadIOData ... 102
PNSWriteIOData ... 102
PORT_CLOSE ... 102
PORT_OPEN ... 102
PORT_READ ... 102
PORT_STATE ... 102
PORT_WRITE ... 102
PPF_COP_COMM ... 101
PUT_ERROR ... 101
PUT_ERROR2 ... 101
READ_AXIS_PAGE ... 103
READ_FILE ... 101
READ_Q_PARAM_* ... 99
READ_RP ... 103
READ_SERC_PARAM ... 99

SAVE_Q_PARAM_* ... 99
SAVE_R_PARAM_* ... 99
SEND_MAIL ... 101
SET_DATE ... 100
SET_SERC_COMMAND ... 99
SET_SERC_PHASE ... 99
SET_TIME ... 100
UBA_ERR_CTRL ... 103
URCV_V ... 98
USEND_V ... 98
UZB_VR ... 103
WRITE_AXIS_PAGE ... 103
WRITE_FILE ... 101
WRITE_PROFILE_INT ... 101
WRITE_PROFILE_REAL ... 101
WRITE_PROFILE_STRING ... 101
WRITE_Q_PARAM_* ... 99
WRITE_RP ... 103
WRITE_SERC_PARAM ... 99
XFIO_CONFIG ... 101
XRIO_STATE ... 101
Funktionsbausteine und Bibliotheken ... 96
Grundinitialisierung ... 81
HALT Betriebszustand ... 92
Handbetrieb ... 88
Hardware-Konfiguration
Einlesen ... 54
I/O-Konfiguration einfügen ... 57
Optionen ... 59
Hochlaufverhalten der SPS ... 83
HyperTerminal ... 123
I/O-Ebene ... 52
I/O-Treiber CANopen ... 69
Inbetriebnahme ... 28, 90
RS232-Schnittstelle ... 123
Initialisierungsdatei ProConOS.ini ... 104
Installation ... 28
AddOns ... 31
MULTIPROG ... 30
OPC-Server ... 30
Schleicher-Dialog ... 32
Systemvoraussetzungen ... 28
Interrupt
Beispiel-POEs ... 63
Eingänge ... 61
Taskstruktur ... 65
UBE 32 0,1I ... 61
Variablen für Verarbeitung ... 61
IP Configuration ... 125
Koppelspeicher
Datenstruktur ... 121
Einfügen ... 48
Hilfe ... 120
nichtremanenter (non retain) Bereich ... 121
remanenter (retain) Bereich ... 121
Übersicht ... 119
Variablen und Tasks ... 119
Versionsauswahl ... 50
Zugriff ... 49, 120
LED-Anzeigen
Ethernet und Sercos III ... 21
Steuerung ... 20
Log-Book ... 91
Log-Datei, von MFA erzeugt ... 85
Memory-Management-Unit ... 11
MFA (Multi Function Application) ... 82
MMU ... 11
Montage ... 14
MULTIPROG
Bibliotheken und Funktionsbausteine ... 96
Einbindung von ProCANopen ... 75
Installation ... 30
Netzwerkvariablen ... 71
Neues Projekt ... 44
Projekt kompilieren und senden ... 46
Projektbaum ... 44
Multi-Task-System ... 107
Netzgeräte ... 26
Netzwerk-Schnittstelle
Computername ... 35
Hausnetz-Adresse ... 34
Inbetriebnahme ... 33
Kommunikationsschema ... 34
Local Area Connection ... 37
PC-Adresse ... 34
PING-Befehl ... 41
Programmierung mit MULTIPROG ... 41
RtOS Virtual Network ... 39
TCP/IP-Einstellungen anpassen ... 36
Textformular ... 40
Vorbereitung ... 33
VxWorks-Adresse ... 34
Windows-Adresse ... 34
Netzwerkvariablen ... 71
NFS and Hardlink ... 129
OPC-Server
Installation ... 30
Koppelspeicherdaten ... 119
PAC ... 11
PDD Process Data Directory ... 80
PDO ... 67
PING-Befehl ... 41
PNSCommunicating ... 102
Prioritäten der Tasks ... 115
ProCANopen
Einbindung in MULTIPROG ... 75
erste Verbindung ... 76
Installation ... 74
Konfiguration mit ... 73
Konfigurationsmanager auswählen ... 78
POE CANdemo ... 79
Test der Netzwerkverbindung ... 79
Process Data Directory ... 80
Process Data Objects ... 67
ProConOS ... 104
ProConOS.ini ... 104
Programm, Zuweisen zu Task ... 117
Programmable Automation Controller ... 11
Programmieren ... 89
Programmierung der SPS ... 92
Projektbaum ... 44
Remote Desktop UltraVNC ... 131
Remote-Zugriff ... 131
Reset-Taster ... 22
RS 232 / RS 422 / RS 485-Schnittstellen ... 19
RTOS Settings ... 130
Schleicher X-Manager ... 125
EWF Configuration ... 128
IP Configuration ... 125
NFS and Hardlink ... 129
RTOS Settings ... 130
Schleicher-Dialog
Installation ... 32
Steuerungsmenü ... 88
Schnittstellen ... 16
CAN ... 19

DVI ... 18
Ethernet ... 17
RS 232 / RS 422 / RS 485 ... 19
Sercos III ... 17
USB 2.0 ... 17
SDO ... 67
sercos III I/O ... 54
Sercos III-Schnittstellen ... 17
Serielle Schnittstelle RS232 ... 123
Service Data Objects ... 67
Sicherheitshinweise ... 7
Software ... 133
SPGs ... *Siehe* Systemprogramme
SPS
Allgemeines Reset ... 94
Betriebssystem ProConOS ... 104
Betriebszustände ... 92
Betriebszustände wechseln ... 93
Grundlagen ... 92
Hochlaufverhalten, Einstellung in MFA ... 83
Programmierung ... 92
Starten/Stoppen der Programmausführung ... 93
Startverhalten nach dem Einschalten ... 94
Steuerung starten/stoppen (MFA) ... 83
Systemvariablen ... 95
Steckplatzreihenfolge ... 14
Steuereinheit ... 15
Abmessungen ... 24
Technische Daten ... 23
Steuerung ausschalten (MFA) ... 83
Steuerung starten (MFA) ... 83
Steuerungsaufbau ... 13
STOP Betriebszustand ... 92
Systemprogramme SPGs ... 110
System-Task ... 110
Systemvariablen der SPS ... 95
Systemvoraussetzungen ... 28
Tabellenverzeichnis ... 136
Task
Anwender-Task ... 108
Default-Task ... 112
Ereignis-Task ... 109
erstellen in MULTIPROG ... 117
Prioritäten ... 115
Prioritätsstufen Übersicht ... 107
SPS/CNC-Synchronisierung ... 119
System-Task ... 110
Task-Information ... 113
Watchdog ... 116
zyklische Task ... 108
TCP/IP-Verbindung ... 28
Technische Daten
Alle Module ... 132
Steuereinheit XCA ... 23
UltraVNC ... 131
USB 2.0-Schnittstellen ... 17
Variablendeklaration
in den Beispielprogrammen der Funktionsbausteine ... 97
Varianten XCA 1100 und XCA 1200 ... 25
VNC ... 131
VxWin ... 11
VxWorks ... 11
Warenzeichenvermerke ... 133
Watchdog ... 116
Webserver ... 80
Windows embedded ... 11
Windows XP embedded ... 125
X1, X2, X3 – Ethernet-Schnittstellen ... 17
X10 – DVI-Schnittstelle ... 18
X11 – CAN-Schnittstellen ... 19
X12 – RS 232 / RS 422 / RS 485-Schnittstellen ... 19
X4, X5 – Sercos III-Schnittstellen ... 17
X6/7, X8/9 – USB 2.0-Schnittstellen ... 17
XUIO-Konfiguration ... 52
Zubehör ... 133
Zugangsberechtigung ... 89
Zuweisen, Programm zu Task ... 117
Zyklische Task ... 108